machines

MDPI

Article

Path Planning for 5-Axis CMM Inspection Considering Path Reuse

Wenzheng Zhao†, Xueqi Wang † and Yinhua Liu *

School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
* Correspondence: liuyinhua@usst.edu.cn
† These authors contributed equally to this work.

Abstract: The 5-axis Coordinate Measuring Machine (CMM) is widely used for quality data collection of the machining parts, such as cylinder blocks and heads of the engines. High efficient inspection path planning for multiple feature groups from different stations is one of the key tasks for CMM application. In engineering practice, the inspection planning of diverse feature groups accounts for large labor cost and process development cycle. To improve the efficiency of path generation for the complex machining part, a five-axis CMM inspection path planning method considering path length, probe rotation and path reusability is proposed. Firstly, the measuring points (MPs) are classified based on feasible inspection direction cone and accessibility of the MPs to achieve the minimum times of probe rotation. Then, the rapidly exploring random trees with multi-root node (RRT-MRNC) algorithm is proposed to implement local path planning considering inspection path reuse. Furthermore, intra-group and inter-group path is generated simultaneously based on the proposed enhanced Genetic Algorithm (GA) algorithm. In order to evaluate the effectiveness of the proposed method, the cylinder block path planning case is used. Compared with the benchmark methods, the total planning time based on the proposed planning method for the dynamic tasks was reduced by 55.2% and 54.9% respectively.

Keywords: path planning; inspection; path reuse; machining product; genetic algorithm; RRT

Citation: Zhao, W.; Wang, X.; Liu, Y. Path Planning for 5-Axis CMM Inspection Considering Path Reuse. *Machines* **2022**, *10*, 973. https://doi.org/10.3390/machines10110973

Academic Editor: Ahmed Abu-Siada

Received: 14 September 2022
Accepted: 10 October 2022
Published: 25 October 2022

1. Introduction

Dimensional measurement of machining products is a key process to achieve product quality monitoring, control, and improvement. The 5-axis CMM with triggering probe is widely used in aerospace, auto body and other fields for its high inspection accuracy, inspection reliability and low long-term cost [1,2]. Machining products with high precision requirements, such as engines and fuselage, etc., require multiple machining stations to complete the production of products. The parts need to be inspected in the critical production stations. Multiple measurements are required for the key MPs of the parts. The MPs in the paper include the curves, sweeping surfaces, hole, etc. In engineering practice, there are different groups of MPs to be inspected for the parts. Therefore, high efficient CMM inspection path planning for different groups is key for high-efficient process development.

However, the existing solution is to use re-planning and teaching or reuse the initial path to achieve the planning of the diverse inspection features groups, which undoubtedly increases the product inspection planning time and labor cost. Therefore, how to reuse the planned local inspection path to meet the need of multi-station inspection planning becomes a critical issue to improve the inspection efficiency of machining products.

1.1. Literature Review

1.1.1. Path Planning Considering Inspection Time

Researchers have conducted numerous studies for shortest path planning with different applications [3,4]. Existing collision-free path planning methods can be divided into

the following three categories: (1) Path planning based on graph search [5,6] (2) Sampling-based path generation strategies [7,8] (3) Machine learning, reinforcement learning-based path planning methods [9]. Han et al. [10] proposed an improved ant colony algorithm to improve the inspection efficiency of CMM. Li et al. [11] constructed an adjacency feature graph for path planning based on Voronoi diagram and designed a greedy search algorithm to search the Voronoi diagram for minimizing the path planning. However, graph-based search methods require modeling of the environment, and the complexity of modeling increases with the complexity of the environment, which will certainly reduce the efficiency of planning.

To solve the above problems, sampling-based path planning method was proposed [12,13]. Slavenko et al. [14] proposed a sampling based strategy to realize collision-free path planning of CMM by simultaneously considering the inspection time and the configuration of the inspection probe. Zhao et al. [15] proposed a geometry-guide method for collision-free probing path generation based on a combination of inspection accuracy and inspection time. However, in the face of complex spatial structures, the sampling-based method is highly prone to path search failure due to the blindness of sampling points. To solve the above problems, existing studies introduce different strategies to improve the effectiveness of sampling and improve the adaptability of the algorithm. Suh et al. [16] proposed an effective nonmyopic path planning algorithm that combined RRT* with the stochastic optimization method cross-entropy to solve the path planning problem in complex space. Lai et al. [17] introduced the Bayesian model within the sampling mechanism to increase the probability of effective sampling in a narrow space, so as to improving the efficiency of planning.

1.1.2. Path Planning Considering Dynamic Tasks

In the face of re-inspection of MPs exceeding the tolerance interval in the machining process, the above path planning method will no longer be suitable due to the computational efficiency of the algorithm and the constraints of manual teaching time. To solve the dynamic path planning problems, many researchers have combined deep learning and reinforcement learning to improve the adaptability of the algorithm and achieve real-time path planning [18,19]. Li et al. [20] quantified the environmental factors of Unmanned Aerial Vehicle and then solved the Unmanned Aerial Vehicle path planning problem based on an enhanced Q-learning algorithm to improve the path quality. Wang et al. [21] proposed a reinforcement learning method based on global guidance for the reprogramming problem in dynamic environments. Li et al. [22] proposed a motion planning framework combining a traditional path planner with deep reinforcement learning to achieve the solution of achieving the optimal path.

Although deep reinforcement learning can solve the dynamic planning problem well, it is difficult to apply to practical engineering for the following two main reasons: (1) It is difficult to provide a large number of real training sample sets in the actual manufacturing process, thus making it difficult for the training model planning results to meet the inspection requirements. (2) Simulation planning results still require manual teaching to eliminate uncertainty of planning results due to product manufacturing error and planning error. Therefore, how to reuse the planning results after the planning and teaching is the key to reducing the amount of teaching and improving the efficiency of inspection.

1.1.3. Path Planning Considering Inspection Accuracy

The inspection accuracy of the machining product is another issue in the inspection process. Existing researches showed that in spite of environmental factors such as temperature and humidity, and frequent orientation changes were the main cause of CMM inspection accuracy degradation [23]. In addition, the orientation of each probe requires calibration of the probe before inspection planning, which greatly increases the calibration time [24]. Zhang et al. [25] proposed a feature-based planning system for CMM inspection processes and determined the minimum number of probe rotations based on a clustering algorithm

and the adjacent redirection matrix of reorientations. However, path optimization is not considered. Li et al. [26] firstly proposed the concept of orientation-point relation shema. Based on the orientation-point relational schema, the minimum orientation algorithm and the shortest path algorithm considering different priorities of probe directions and path length was proposed. However, they used a greedy algorithm to plan the path between different sets of categorical features, resulting in planning results are difficult to achieve optimality. In addition, some scholars have proposed a knowledge-based inspection path planning system, which combines human experience with commercial software,such as UG, PC-DMIS, to achieve optimization of probe direction and path length [27,28]. However, those methods still cannot achieve automatic path planning.

1.1.4. Contribution of This Paper

The state-of-the-art studies are mainly to find a collision-free path that considers inspection time or inspection accuracy. However, few studies have considered the impact of path reuse on efficiency improvement. In machining and other manufacturing scenarios, the key features are often need to be measured several times to ensure the accuracy of the product. In addition, the key features to be inspected in each process, such as launch and mass production of the machining part, are different. To solve the above problems, the following two strategies are proposed: (1) The initial path reuse; (2) The new path re-planning. For the former strategy, there will be a large number of redundant paths. For the latter one, though the generation of redundant paths are reduced but the teaching time for the new path is increased. Both of the above strategies lead to a reduction in CMM inspection efficiency. Therefore, it is necessary to develop a path reuse strategy that also accounts for the number of path and probe measurement directions to reduce the time for repeated teaching, and improve inspection efficiency. In this paper, a framework for inspection path planning considering path reuse is established by considering the number of probe orientations, inspection time and reusability of inspection path, the main contributions include the following.

(1) Based on the feasible measurement orientation cone and accessibility, the set covering problem model for MPs classification is established, and the minimum coverage cluster is solved based on greedy algorithm.
(2) The rapidly exploring random trees with multi-root node competition algorithm is proposed to implement inspection path planning considering inspection path reuse.
(3) The global path planning problem with multiple measuring points clusters is transformed into a Generalized Traveling Salesman Problem (GTSP) model and solved based on an enhanced Genetic Algorithm (GA) algorithm.

The rest of this article is organized as follows. In Section 2, an inspection path planning framework for the machining products is established by considering the number of probe orientations, inspection time and reusability of inspection path. In Section 3, an RRT-MRNC algorithm is proposed to optimize the inspection path, and an enhanced genetic algorithm is used to solve the global path planning problem with multiple MP clusters. Experiments are implemented in Section 4. Section 5 concludes the article.

2. Problem Statement

2.1. Problem Description

In the machining process, Figure 1 shows the cylinder block inspection station, critical MPs or MPs with dimensions exceeding the tolerance interval need to be measured several times in a specific process. If the set of MPs sequence for collision-free inspection path a process is π_i', and when some MPs in the set π_i' is out of the tolerance interval, it still needs to be measured again in a specific process. To eliminate the re-planning and teaching time, we propose an inspection planning method based on the reuse of planning path. That is, without increasing the obstacle avoidance points, some of MPs and obstacle avoidance points in the π_i' planning results are deleted, and the retained planning results still meet

the requirements of collision-free. Overall, the following three aspects need to be taken into account:

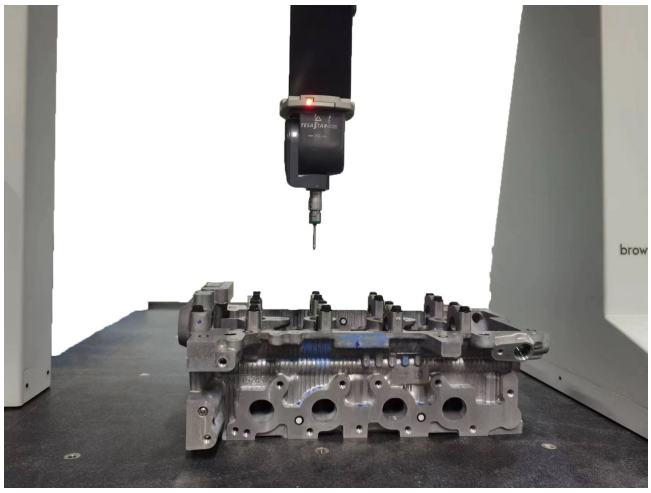


Figure 1. Cylinder block inspection station.

(1) Probe tolerability: The frequent probe rotations not only increase the inspection time, but also increase the inspection error. Therefore, the number of probe rotations need to be used as the optimization objective to classify the MPs.
(2) Path reusability: For the initial planned collision-free path, the planned path result is still a collision-free path after deleting some of MPs and obstacle avoidance points.
(3) Inspection efficiency: For the classified MPs, the inspection time of in-group and inter-group MPs need to be optimized.

2.2. Path Reuse Planning Optimization Modeling

The path planning problem in Section 2.1 can be abstracted into the following two problems: (1) The global path planning problem with multiple MPs clusters, which can be transformed into a GTSP model. Figure 2 gives a schematic diagram of the GTSP; (2) The initial collision-free path reuse problem, i.e., path planning for dynamic tasks without adding obstacle avoidance points to the initial path, can be solved as a dynamic traveling salesman problem with constraints. Let $\mathbf{S}$ be the set of n classified sets obtained by optimizing the number of probe orientation, $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$, where each set $\mathbf{S}_i \in \mathbf{S}$ consists of m_i MPs $\mathbf{S}_i = \{\mathbf{Q}_{i,1}, \ldots, \mathbf{Q}_{i,m_i}\}$. The purpose of the first problem is to determine the sequence of the MPs $\lambda = \{\lambda_1, \ldots, \lambda_{2n}\}$ to the sets $\mathbf{S}_{\lambda_i} \in \mathbf{S}$ together with the relative MPs positions $\pi = \{\mathbf{p}_1, \ldots, \mathbf{p}_{2n}\}$ such that $\mathbf{p}_i \in \mathbf{S}_i$, and the number of intersections of π with any $\mathbf{S}_i$ is always 2, which can be expressed as $\forall\, |\pi \bigcap \mathbf{S}_i| = 2$, where $|\cdot|$ is an operator that calculates the number of elements in the intersection of two sets. Furthermore, The second problem is to determine a collision-free path $\pi_i' = \{\mathbf{q}_{i,1}, \ldots, \mathbf{q}_{i,m_i}\}$ of each $\mathbf{S}_i$, where $\mathbf{q}_{i,1} = \mathbf{p}_i$, $\mathbf{q}_{i,m_i} = \mathbf{p}_{i+1}$.To optimize the path planning problem, an optimization framework is proposed with the objective function shown in Equation (1):

$$\min_{\lambda,\pi,\pi'} \mathbf{L} = \sum_{i=1}^{n-1} \left(\left\| (\mathbf{p}_{\lambda_i}, \mathbf{p}_{\lambda_{i+1}}) \right\| + \sum_{k=1}^{m_i} \left\| (\mathbf{q}_{i,k}, \mathbf{q}_{i,k+1}) \right\| \right) + \left\| (\mathbf{p}_{\lambda_n}, \mathbf{p}_{\lambda_1}) \right\| \tag{1}$$

$$\text{s.t.} \begin{cases} \lambda = \{\lambda_1, \ldots, \lambda_{2n}\}, 1 \le \lambda_i \le n, \lambda_i \ne \lambda_j \; for \; i \ne j, \\ \pi = \{\mathbf{p}_1, \ldots, \mathbf{p}_{2n}\}, \\ \forall\, |\pi \bigcap \mathbf{S}_i| = 2, i \in \{1, \ldots, n\}, \\ \pi_i' = \{\mathbf{q}_{1,1}, \ldots, \mathbf{q}_{i,m_i}\},\ \mathbf{q}_{i,1} = \mathbf{p}_i,\ \ \mathbf{q}_{i,m_i} = \mathbf{p}_{i+1}, \\ \pi' = \{\pi_1', \ldots, \pi_n'\}, \\ \mathbf{G}(\mathbf{q}_{i,k}, \mathbf{q}_{i,l}) \cap \boldsymbol{V} = \emptyset,\ k, l = 1, \ldots, m_i,\ k \ne l \end{cases} \tag{2}$$

where $\mathbf{G}(\mathbf{q}_{i,k}, \mathbf{q}_{i,l})$ is denotes the geometric space occupied by the CMM moving from the MP $\mathbf{q}_{i,k}$ through the obstacle avoidance point to the MP $\mathbf{q}_{i,l}$; $\mathbf{V}$ is denotes the geometric space occupied by the inspected surface, including the inspected free-form part, tooling bracket system, etc.; π' denotes the set of all $\mathbf{S}_i$ collision-free path.

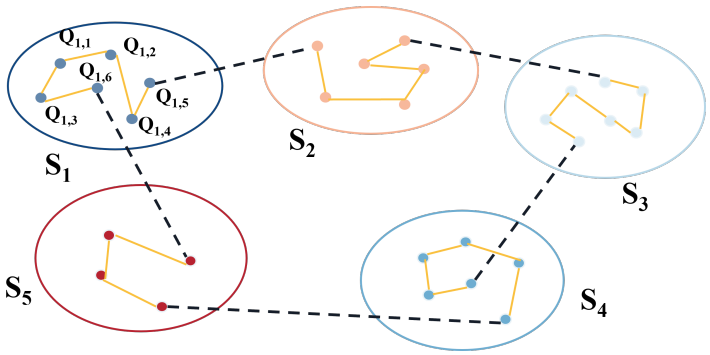


Figure 2. Schematic diagram of GTSP problem.

In addition, without loss of generality, The following assumptions are listed:

(1) The 5-axis CMM consists of a rotating 2-axis probe and a traditional moving 3-axis. During the inspection process, only movement or rotation is executed at one time.
(2) During the measurement process, the angle between the vector direction of the probe and the normal direction of the feature should meet the constraints of the measurement specification to ensure measurement accuracy.
(3) The velocity and acceleration of the CMM are assumed to be constant throughout the inspection process.

3. Path Planning Considering Path Reuse

3.1. Measuring Points Classification and Probe Matching

The frequent probe rotation will increase the inspection time and inspection error. To improve the tolerability of probe, a valid probe orientation should be satisfied the following two criteria: (1) The inspection of features is achieved with a minimum number of probe orientations while meeting the measurement specifications. (2) Probe approaches MP without any collision. In order to obtain the minimum number of probe rotations, we firstly build a feasible measuring orientation cone based on the inspection features. In practice, the probe measures each MP in a range of acceptable orientation kept between $(-\theta, \theta)$, where θ is specified in the CMM probe user manual. A measuring point and its feasible measurement orientation cone are given in Figure 3, and the feasible vector set $\mathbf{V}_i$ of the MP $\mathbf{Q}_i$ can be calculated by [26].

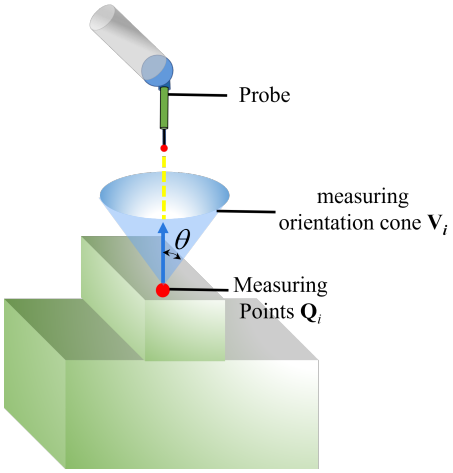


Figure 3. Schematic diagram of measuring orientation cone.

Feature classification results are the determining factor for the success of path reuse planning. On the premise of satisfying the above two specifications, it is assumed in this paper that all MPs are distributed to at most K ($K \leq 2$) surfaces, which is a sufficient condition to achieve path reuse, as shown in Figure 4a,b. When the MPs are distributed to more than 2 surfaces, a large number of redundant inspection path will be generated to meet the dynamic planning requirements, leading to an increase in inspection path length.

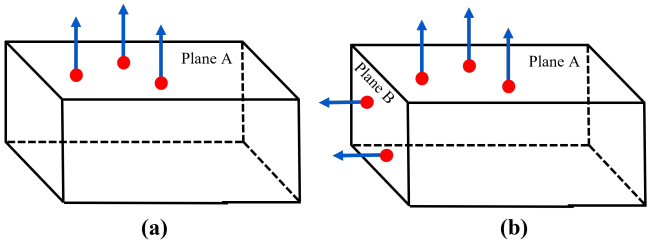


Figure 4. Distribution of MPs. (**a**) The MPs are distributed in the same plane (**b**) The MPs are distributed in adjacent planes.

Therefore, the MPs classification and probe matching should meet following constraint , and the greedy algorithm is used to optimize the number of the probe orientation, the related solution method can be found in [29].

$$\text{s.t.} \begin{cases} < \mathbf{V}_i, \mathbf{V}_{probe,j} > < \theta_i \\ \mathbf{G}_{\mathbf{Q}_i} \cap \boldsymbol{V} = \emptyset \\ \mathbf{Q}_1 \oplus \mathbf{Q}_2 \oplus \ldots \oplus \mathbf{Q}_{\mathbf{m}_i} \leq 2 \end{cases} \tag{3}$$

In the above constraint , $\mathbf{V}_{probe,j}$ represents the jth orientation of probe; $< \mathbf{V}_i, \mathbf{V}_{probe,j} >$ denotes the clamp angle between the MPs $\mathbf{Q}_i$ and the jth orientation of probe; $\mathbf{G}_{\mathbf{Q}_i}$ is the geometric space occupied by the probe; $\oplus$ denotes the operator symbol for calculating the number of planes in which MPs are located.

Based on the aforementioned steps, the minimum MP sets $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$ can be obtained, which provide the optimal probe orientation for subsequent path planning and path reuse.

3.2. Inspection Path Reuse Based on RRT-MRNC

To ensure the reusability of the planned path of the classified MP sets, the following two requirements must be met during path planning: (1) The planned path are the shortest collision-free path; (2) no collision occurs between any two MP path. To address the above problems, we propose the RRT-MRNC algorithm to meet these criteria simultaneously. The RRT algorithm is an incremental method to avoid modeling the space by performing collision detection on sampled points in the state space, which can search the high-dimensional space quickly and efficiently and is suitable for path planning problems in complex dynamic environments [30].

To formulate the inspection path planning and reuse problem clearly, we introduced the following symbol definitions, $\mathbf{G}_{free}$ denotes the free space; $\mathbf{Q}_{i,k}$ and $\mathbf{x}_{i,start}$ denotes the k^{th} MP position and start position in the ith classified set, respectively; $\mathbf{T.V}_{i,k}$ and $\mathbf{T.E}_{i,k}$ are denotes as the set of vertices of the tree edges relating to the kth MP in the ith classification set, respectively; $\mathbf{T.V}_i = \{\mathbf{T.V}_{i,1}, \ldots, \mathbf{T.V}_{i,m_i}\}$ and $\mathbf{T.E}_i = \{\mathbf{T.E}_{i,1}, \ldots, \mathbf{T.E}_{i,m_i}\}$. The proposed RRT-MRNC algorithm generally follows three steps: firstly, the new candidate tree nodes $\mathbf{x}'_{init_{i,k}}$ is determined based on the multi-objective constraints by using the proposed **RootSampling**() function (Algorithm 2). Secondly, the optimal tree roots $\mathbf{x}_{init_{i,k}}$ are selected based on the cost function (Equation (4)) and the proposed **RootRewire**() function (Algorithm 4). Thirdly, local collision-free paths from the root of the tree $\mathbf{x}_{init_{i,k}}$ to $\mathbf{Q}_{i,k}$ generated based on the traditional RRT* search strategy. The algorithm terminates when all the collision-free path is generated. The overviews of the proposed RRT-MRNC

algorithm which generate the local collision-free path from the $\mathbf{x}_{i,start}$ to each $\mathbf{Q}_{i,k}$ in $\mathbf{S}_i$ are given in Algorithm 1.

Algorithm 1 Local path planning based on RRT-MRNC algorithm

$\mathbf{T}.\mathbf{E}_i = \emptyset$;
$\mathbf{T}.\mathbf{V}_i = \emptyset$;
$\mathbf{x}'_{init,i} \leftarrow$ **RootSampling**$(\mathbf{S}_i, \mathbf{G}_{free})$;
j=0;
while j<$Root_{max}$ **do**
  $\mathbf{x}_{init,i} \leftarrow$ **RootRewire**$(\mathbf{x}'_{init,i}, \mathbf{S}_i, \mathbf{G}_{free})$;
  j=$j+1$;
end while
for $k = 1 : \mathbf{m}_i$ **do**
  $\mathbf{T}.\mathbf{V}_{i,k} \leftarrow \mathbf{Q}_{i,k}$;
  $\mathbf{T}.\mathbf{V}_{i,k} \leftarrow \mathbf{x}_{i,start}$;
  $\mathbf{T}.\mathbf{E}_{i,k} \leftarrow \emptyset$;
  $\mathbf{T}.\mathbf{V}_{i,k} = \mathbf{T}.\mathbf{V}_{i,k} \cup (\mathbf{x}_{init_{i,k}})$;
  $\mathbf{T}.\mathbf{E}_{i,k} = \mathbf{T}.\mathbf{E}_{i,k} \cup (\mathbf{x}_{init_{i,k}}, \mathbf{x}_{i,start})$;
  $\{\mathbf{T}.\mathbf{V}_{i,k}, \mathbf{T}.\mathbf{E}_{i,k}\} \leftarrow$ **RRT***$(\mathbf{T}.\mathbf{V}_{i,k}, \mathbf{T}.\mathbf{E}_{i,k})$;
  $\mathbf{T}.\mathbf{E}_i \leftarrow \mathbf{T}.\mathbf{E}_{i,k}$;
  $\mathbf{T}.\mathbf{V}_i \leftarrow \mathbf{T}.\mathbf{V}_{i,k}$;
end for
return $\mathbf{T}.\mathbf{V}_i, \mathbf{T}.\mathbf{E}_i$

In Algorithm 1, $Root_{max}$ is the maximum number of the **RootRewire**() iteration; $\mathbf{x}'_{init_{i,k}}$ denotes the candidate root node of the kth MP in $\mathbf{S}_i$; $\mathbf{x}_{init_{i,k}}$ denotes the optimal root node of the kth MP in $\mathbf{S_i}$; $\mathbf{x}'_{init,i}$ represent the sets of all the candidate tree nodes, that is $\mathbf{x}'_{init,i} = \{\mathbf{x}'_{init_{i,1}}, \ldots, \mathbf{x}'_{init_{i,m_i}}\}$; $\mathbf{x}_{init,i}$ represent the set of all the optimal tree nodes, that is $\mathbf{x}_{init,i} = \{\mathbf{x}_{init_{i,1}}, \ldots, \mathbf{x}_{init_{i,m_i}}\}$.

We randomly determine multiple tree root nodes in the safe space, and the generated root nodes can be obstacle avoidance points for multiple MPs, and further perform local path planning for each MP based on these root nodes. Furthermore, the following three aspects need to be satisfied when searching for the root nodes:

(1) Without adding avoidance points, all paths between root nodes and from root nodes to the start are collision-free.
(2) The determined root node must be within the safe space.
(3) Ignore collision between root nodes and MP path.

The purpose of path reuse is to ensure that there is still no conflict between rejoined path after deleting obstacle avoidance points. Therefore, we elaborately designed the **RootSampling**() to meet this goal. Specifically, for some MPs are required re-inspected, we only need to delete the relevant tree nodes between the MPs which do not need to be re-inspected and the tree root nodes, then connect the disconnected path to complete the re-planning, Figure 5 shows the path reuse schematic. The pseudocode of the proposed **RootSampling**() function is shown in Algorithm 2.

In Algorithm 2, *NotRootNodes* is the flag to determine whether the algorithm is terminated, which returns 0 if all tree nodes have been found, and 1 otherwise; **iscollision**() is a function to evaluate whether the CMM collides with the static environment when moving between two MPs, the related collision detection algorithm can be seen in the [3]. If the collision occurs, it is *True*, otherwise, it is *False*; **SearchingforRoots**() is a procedure to search for the candidate tree nodes $\mathbf{x}'_{init,i}$.

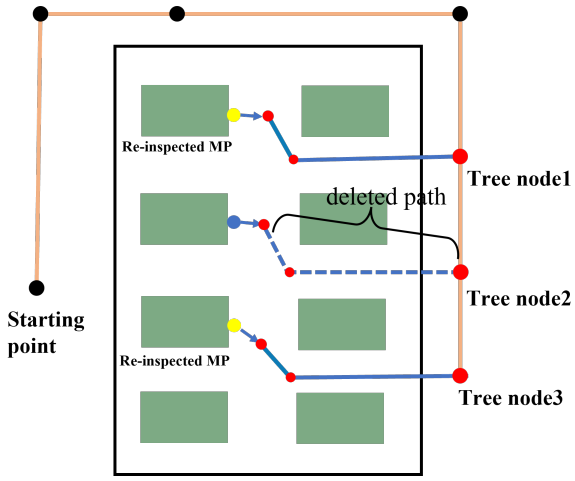


Figure 5. Path Reuse Schematic.

Algorithm 2 Candidate tree nodes $\mathbf{x}'_{init,i}$ determination

Function: $\mathbf{x}'_{init,i} \leftarrow$ **RootSampling**$(\mathbf{S}_i, \mathbf{G}_{free})$
$NotRootNodes = 1$;
$\mathbf{x}'_{init,i} = \emptyset$;
while $NotRootNodes = 1$ **do**
  for $\mathbf{Q}_{i,k} \in \mathbf{S}_i$ **do**
    $\mathbf{x}'_{init_{i,k}} =$ **SearchingforRoots**$(\mathbf{Q}_{i,k}, \mathbf{G}_{free})$;
    $\mathbf{x}'_{init,i} = \mathbf{x}'_{init,i} \bigcup \mathbf{x}'_{init,i,k}$
  end for
  if iscollision$(\mathbf{x}'_{init,i}, \mathbf{x}_{i,start}) = False$ **then**
    for init$_q \in \mathbf{x}'_{init,i}$ & **init**$_p \in \mathbf{x}'_{init,i}$ & **init**$_p \neq$ **init**$_q$ **do**
      if iscollision$(\mathbf{init}_q, \mathbf{init}_p) = true$ **then**
        $NotRootNodes = 1$;
        break
      else
        $NotRootNodes = 0$;
      end if
    end for
  end if
end while
return $\mathbf{x}'_{init,i}$

Furthermore, we deliberately create the **SearchingforRoots**() function to improve the efficacy of sampling, the pseufocode is shown in Algorithm 3, where **CreateSafetySpace**() is a function to determine the safe space $\mathbf{G}_{safety}$ and the range of the safe space *Range*, the safe space build-up can be referred to [6]; **CreateVector**() is a function that creates a frequently used vector library **vector**, which includes vector directions of MPs in $\mathbf{Q}_{i,k}$, the vector contained between the two MP vectors, etc.; ε_0 is the threshold to judge whether to select the vector direction in **vector**; *rand* denotes generating a random number between 0 and 1; ζ denotes the amount of range disturbance.

The clustered MPs in Section 3.1 exist in at most two adjacent faces with the angle between MPs less than θ_i. Furthermore, the root node can typically be found for clustered MPs that only exists in a single plane by conducting a search along the vector direction of the MPs, as shown in the Figure 4a. Similarity, for the clustered MPs that exist on two adjacent faces, a search along the direction between the vectors of two MPs usually determines the root node, As shown in the Figure 4b. To boost the efficiency of root node searching, we sample the root node by using the MPs as the center and the established vector library as the search direction. Besides, in order to improve the randomness of

sampling, we add the probability threshold ε_0, when $rand() \geq \varepsilon_0$, the sampling point can be searched in any direction, and ε_0 should have a value slightly greater than 0.5 to speed up the search.

Moreover, in order to ensure that the above sampling method based on search direction can collect all points in $\mathbf{G}_{safety}$ space, the sampling perturbation amount ζ is added in this paper, that is, the full sampling in $\mathbf{G}_{safety}$ space can be achieved by the combination of expanding the sampling range, and the value of ζ can be slightly larger than the shortest distance from each MP to $\mathbf{G}_{safety}$.

Algorithm 3 Searching for candidate tree nodes $\mathbf{x}'_{init_{i,k}}$

Function: $\mathbf{x}'_{init_{i,k}} = \textbf{SearchingforRoots}(\mathbf{Q}_{i,k}, \mathbf{G}_{free})$
$(\mathbf{G}_{safety}, Range) = \textbf{CreateSafetySpace}(\mathbf{Q}_{i,k}, \mathbf{G}_{free})$
$\textbf{vector} = \textbf{CreateVector}(\mathbf{Q}_{i,k})$
if $rand() < \varepsilon_0$ **then**
  $vec = \textbf{SelectVector}(\textbf{vector})$;
else
  $vec = rand()$;
end if
while $x \notin \mathbf{G}_{safety}$ **do**
  $x = x + vec \cdot rand() \cdot Range + \zeta$
end while
$\mathbf{x}'_{init_{i,k}} = x$;
return $\mathbf{x}'_{init_{i,k}}$

The length of the inspection path is significantly influenced by the position of the root node, and we use the root node rewiring method to optimize the root node. The procedure of **RootRewire**() is shown in Algorithm 4, where $\textbf{cost}_{min}$ is the current minimum cost calculated according to Equation (4); **CalculateCost**() denotes the distance objective function using Equation (4).

$$\textbf{cost}_{min} = \left\|(Q_{i,k}, x_{\text{init},i}(k))\right\| + \left\|(x_{i,\text{ start}}, x_{\text{init},i}(k))\right\| \tag{4}$$

Algorithm 4 Determining the optimal tree nodes $\mathbf{x}_{init,i}$

Function: $\mathbf{x}_{init,i} \leftarrow \textbf{RootRewire}(\mathbf{x}'_{init,i}, \mathbf{S}_i, \mathbf{G}_{free})$
$\mathbf{x}_{init,i} = \emptyset$;
for $k = 1 : m_i$ **do**
  $\mathbf{x}'_{init_{i,k}} = \mathbf{x}'_{init,i}(k)$
  $\mathbf{Q}_{i,k} = \mathbf{S}_i(k)$;
  $\textbf{cost}_{min} = \textbf{CalculateCost}(\mathbf{x}'_{init_{i,k}})$;
  $\mathbf{x}'_{new,i,k} = \textbf{SearchingforRoots}(\mathbf{Q}_{i,k}, \mathbf{G}_{free})$
  $\textbf{cost}_{new} = \textbf{CalculateCost}(\mathbf{x}'_{new,i,k})$;
  if $\textbf{cost}_{new} < \textbf{cost}_{min}$ **then**
    $\textbf{cost}_{min} = \textbf{cost}_{new}$;
    $\mathbf{x}_{init,i} = \mathbf{x}_{init,i} - \mathbf{x}'_{init_{i,k}}$;
    $\mathbf{x}_{init,i} = \mathbf{x}_{init,i} \cup \mathbf{x}'_{new,i,k}$;
  end if
end for
return $\mathbf{x}_{init,i}$

Based on the optimal root node $\mathbf{x}_{init,i}$, we need to further optimize the path between $\mathbf{x}_{init,i,k}$ and MP $\mathbf{Q}_{i,k}$. The complete local collision-free path can be generated based on the RRT* algorithm, and the related algorithm can be seen in [31].

Based on the above RRT-MRNC algorithm, the path from any MP to the starting point within a given classification set $\mathbf{S}_i$, and the path of any two MPs connected by the root node is collision-free can be determined.

3.3. Global Path Planning Based on Enhanced Genetic Algorithm

For the local collision-free path based on the Section 3.2, we need to further perform global path optimization to ensure the improvement of inspection efficiency. The global path planning including the following three problems: (1) The collision-free path between different sets needs to be determined (2) The measurement sequence between each classified set needs to be determined, which is a GTSP problem (3) The intra-group MPs sequential optimization, which is a TSP problem. For the first sub-problem, it can be solved based on Algorithm 1. The latter two problems are both NP-hard problem, and methods for solving such problems include branch and bound mixed-integer linear programming and heuristic algorithms [3]. In this paper, an enhanced genetic algorithm is used to solve the above two sub-problems simultaneously to achieve the optimal solution of global path planning.

The genetic algorithm achieves population optimization by three operations on population individuals: selection, crossover, and mutation. The genetic algorithm has significant evolutionary ability in dealing with an NP-hard problem. However, it is prone to local optimum in solutions. To tackle this problem, heuristic mutation [32] and 2-opt operation are introduced in this paper to achieve the global path optimization solution. To better introduce the proposed enhanced GA algorithm, we use *gene* to denote the encoded bit string number of each MP. A gene sequence constitutes the *chromosome*, which contains the measurement sequence information of the MP. The solutions generated from the chromosome is named with *individual*. *Population* represents a collection of chromosomes. The encoding in the chromosome that represents the MPs in each set of $\mathbf{S}_i$ are called *sub-segments*. The flowchart of the enhanced GA is summarized in Algorithm 5.

In the Algorithm 5, N_{pop} is the number of initial populations; P_c represent the crossover probabilities; P_m represent the mutation probabilities; T_{iter} is the maximum number of iterations of the algorithm; **Encoding**() generates a set of chromosomes (population), which includes many sequencing schemes; **Decoding**() decodes the chromosome to obtain the sequence of global planning; C is the population obtained by **Encoding**(); **Fitness Evaluation**() denotes the evaluation function of the individual, the detailed calculation process is shown in Equation (5); C_{elite}, $C_{non\text{-}elite}$ represent the elite chromosomes or the non-elite chromosomes selected from the fitness function **Fitness Evaluation**(), respectively; **Choose**() ranks individuals from smallest to largest according to the fitness function and selects the top N_e elite chromosome; **Crossover**() denotes the function of two chromosomes crossing over to generate two new chromosomes, and the detailed steps can be found in [33]; **Mutation**() represents the function that selects and modifies a sub-segment of a chromosome to obtain a new chromosome; **2-opt**() represents the function that selects two positions in a chromosome sub-segment to be swapped to obtain a new chromosome; $\pi'_{new,i}$, π_{new} denote the set of all newly generated collision-free path sequences of $\mathbf{S}_i$ and the result of all inter-group sequences, respectively.

The chromosomes are integer encoded to decrease the complexity of encoding/decoding, Figure 6, Figure 7 represent the process of encoding and decoding respectively. As shown in the Figure 6, with each encoding denoting the index value of a MP, before and after each vertical line in the figure indicates the ranking result of different $\mathbf{S}_i$. The inspection sequence will be generated by decoding. As shown in Figure 7, the gene values in each sub-segment were firstly ranked from smallest to largest to determine the sequence π'_i of each $\mathbf{S}_i$. Secondly, the gene with the smallest value in each sub-segment was selected for ranking to determine the sequence π between each $\mathbf{S}_i$.

Algorithm 5 Optimize global path via enhanced GA

Function: $(\pi_i', \pi) =$ **Enhanced GA**$(\mathbf{S}, N_{pop}, P_c, P_m, T_{inter}, k = 1)$
$C =$ **Encoding**$(\mathbf{S}, N_{pop})$;
$(\pi', \pi) =$ **Decoding**(C);
$Value =$ **Fitness Evaluation**(π_i', π);
while $k < T_{iter}$ **do**
  $(C_{Non-elite}, C_{elite}) =$ **Choose**$(C, Value)$;
  $C_{new} =$ **Crossover**$(C_{Non-elite}, C_{elite}, P_c)$;
  $C_{new} =$ **Mutation**(C_{new}, P_m);
  $C_{new} =$ **2-opt**(C_{new});
  $(\pi'_{new,i}, \pi_{new}) =$ **Decoding**(C_{new});
  $Value_{new} =$ **Fitness Evaluation**(π'_{new}, π_{new});
  if $Value_{new} - Value < 0$ **then**
    $\pi' = \pi'_{new}$;
    $\pi = \pi_{new}$;
  end if
  $k = k + 1$
end while
return π', π

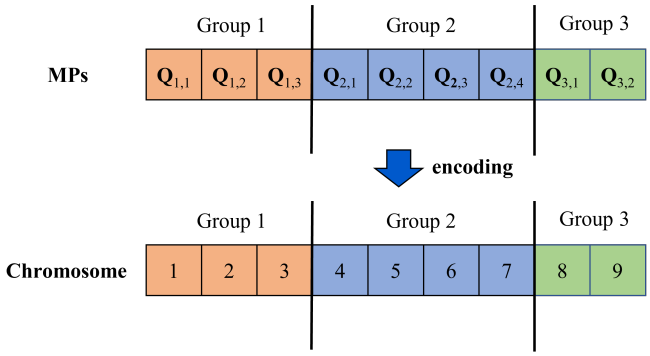


Figure 6. Schematic of the encoding process.

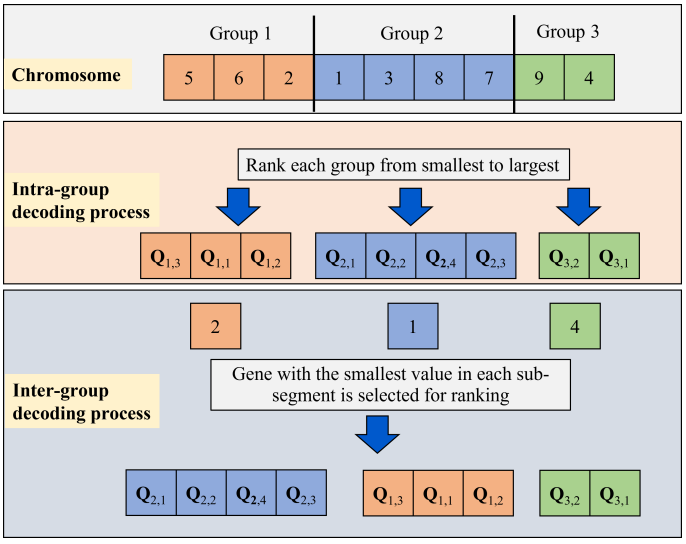


Figure 7. Schematic of the decoding process.

The fitness function is a criterion for evaluating the merit of chromosomes, and a reasonable choice of the fitness function is crucial in the process of algorithm execution, the fitness function can be established as Equation (5).

$$Value(\pi,\pi') = \sum_{i=1}^{n-1}\left(\|(\pi(i),\pi(i+1))\| + \sum_{k=1}^{m_i-1}\left\|\left(\pi_i'(k),\pi_i'(k+1)\right)\right\| + \left\|\left(\pi_i'(1),\pi_i'(k)\right)\right\|\right) + \|(\pi(n),\pi(1))\| \quad (5)$$

In the fitness function, we use π, π' to denote an individual which contains the inspection sequence of each $\mathbf{S_i}$ and the inspection sequence of MPs in each $\mathbf{S_i}$, respectively. The value of the fitness function is the smaller, the better.

The mutation operation enriches the diversity of the population and increases the probability of the optimal solution being searched, heuristic mutation and 2-opt operations are introduced to optimize the overall path and the inter-group path. A schematic diagram of the heuristic operator is shown in Figure 8, which operates in detail as follows:

(1) A new individual can be determined by randomly selecting X positions of encoding from the individuals and arbitrarily swapping the positions with each other.
(2) Calculate the fitness function of $X! - 1$ new individuals and select the individual with the smallest fitness as the new individual.

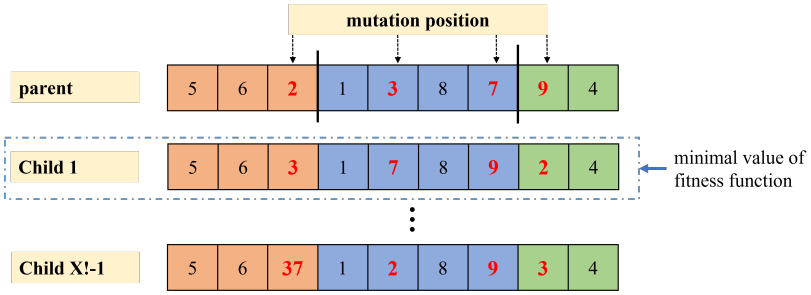


Figure 8. Schematic diagram of the heuristic operator.

The algorithm is easily trapped in the local optima caused by the coupling effect between inter-group sequential optimization and intra-group sequential optimization. Thus, the 2-opt is applied to the mutation operation of genetic algorithm to get rid of local optimization, as shown in Figure 9, the specific steps are as follows:

(1) The genes with the maximum and minimum values in any sub-segment of the chromosome are randomly selected, and the positions are exchanged between each other to determine the new individual.
(2) If the fitness function of the newly generated individual is smaller than the previous one, then use the newly generated individual as a new individual. Otherwise, go to step1.

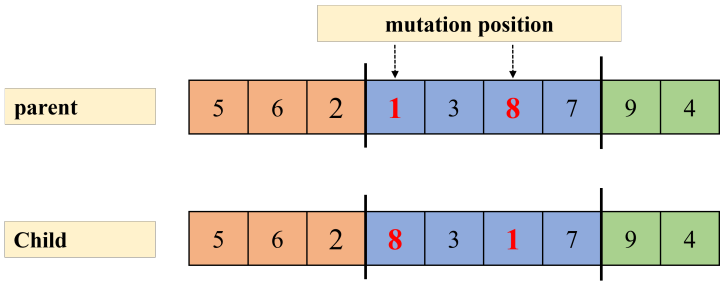


Figure 9. Schematic diagram of the heuristic operator.

4. Experiment and Discussion

In this section, the proposed method is validated for CMM path planning on the cylinder block, the structure of the cylinder block is shown in Figure 10, with 39 MPs on its surface, which requires a 5-axis CMM to realize the inspection of all the MPs within different

processes. Besides, to better verify the effectiveness of the proposed method, two state-of-the-art path planning method, a dynamic searching volume-based(DSV) algorithm [3] and a measuring safety space (MSS) method [6] , were used as benchmark methods for comparative analysis. The above cylinder block was inspected on a HEXAGON GLOBALPLUS CMM. The measurement range is 700 × 1000 × 700 mm, the maximum allowable value error is 1.7 + 3.0 L/1000 µm and the maximum allowable measurement error is 2.3 µm for this CMM. The dimension of the probe used for inspection are shown in Figure 11, and the related parameters of CMM and probe are shown in Table 1. The rotation and movement speeds in the teaching stage are set as 0.47 mm/s and 40 mm/s, respectively.

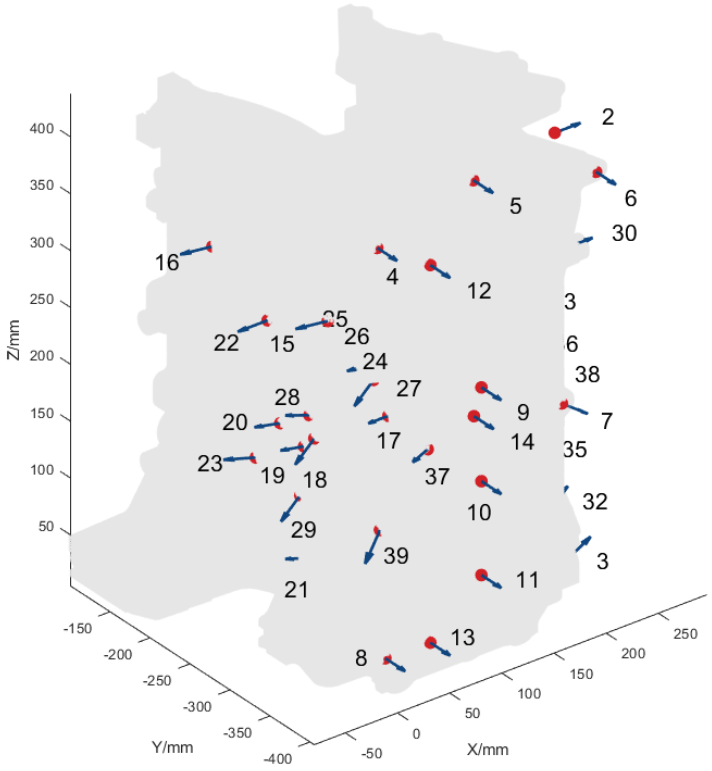


Figure 10. MP layout on the cylinder block.

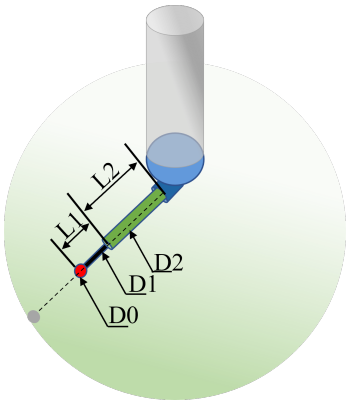


Figure 11. Probe model.

Table 1. Parameters of the CMM and probe.

Parameters	L1 (mm)	L2 (mm)	D0 (mm)	D1 (mm)	D2 (mm)	Rotation Speed (mm/s)	Movement Speed (mm/s)	MP Inspection Time (s)
Value	50	150	3	3	11	1	85	1

4.1. Comparative Analysis and Verification

Based on the MP classification and probe matching in Section 3.1, the MP grouping results and the accessible angle of the probe can be determined, and the results of the subgroups are shown in Table 2. The maximum probe rotation angle for each $\mathbf{S}_i$ is 35.288°, 4.744°, 84.379°, 54.050°, 30.616°. The maximum probe rotation in all groups is less than 90°, which effectively reduces the measurement time caused by probe rotation and also reduces the loss of accuracy caused by frequent probe rotation. In addition, the MPs in each $\mathbf{S}_i$ group are distributed in two planes at most, and the two planes are adjacent to each other, which provides the necessary condition for path reuse.

Table 2. MP classification results.

Subgroup	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$	$\mathbf{S}_5$
MP No.	{1, 2, 3, 30, 32, 33, 34, 36, 38}	{7,31,35}	{18,27,29,39,8,9,10,11,12,13,14}	{4,5,6,26,28,27}	{15,16,17,19,20,21,23,24,25,37}

As the sequential planning of groups are not considered in the comparison method, and the proposed enhanced GA algorithm is used to optimize the sequence between groups in the comparison method to ensure the fairness of the planning results. Figure 12 shown the multi-group sequential planning diagram based on the enhanced GA algorithm, respectively. From the planning sequence results, it can be seen from Figure 12 that there is no "crossover" phenomenon within and between groups, which can effectively improve the inspection time.

The inspection path planning of CMM generated by DSV algorithm, MSS algorithm and RRT-MRNC algorithm are shown in Figures 13, Figure 14 and Figure 15, respectively, where the black star represents the starting position, the green dot represents the obstacle avoidance point, the red dot represents the MP, the colored lines represent path. Table 3 shows the inspection time and computation time by different methods. In Table 3, the motion time represents the sum of the movement time, the rotation time and the inspection time. Total time represents the sum of the computation time, the motion time and the teaching time spent in planning a given set of tasks. The probe rotation time based on the proposed method is reduced by about 17.9%, 51.3%, compared to the DSV and MSS algorithms. Compared to DSV algorithm, the motion time and teaching time determined by the proposed RRT-MRNC method are increased by 12.3% and 38.8%, and decreased by 29.0%, 17.3%, respectively, compared to the MSS algorithm. The RRT-MRNC algorithm ensures that the planned path remains collision-free after removing some MPs and related obstacle avoidance points by adding redundant paths. Thereby, the motion time and teaching time are slightly longer than the DSV methods. However, the computation time and total time based on the proposed RRT-MRNC method are lower than the benchmark method, and the total time is reduced by 27.0% and 41.4% compared to DSV and MSS, respectively.

Table 3. Comparison of planning results by different methods.

	Computation Time (s)	Probe Movement Time (s)	Probe Rotation Time (s)	Inspection Time (s)	Motion Time (s)	Teaching Time (s)	Total Time (s)
DSV	935.7	122.0	117.0	39.0	278.0	546.9	1760.5
MSS	864.1	203.7	197.0	39.0	439.7	890.5	2194.3
RRT-MRNC	213.2	177.4	96.0	39.0	312.4	759.3	1285.0

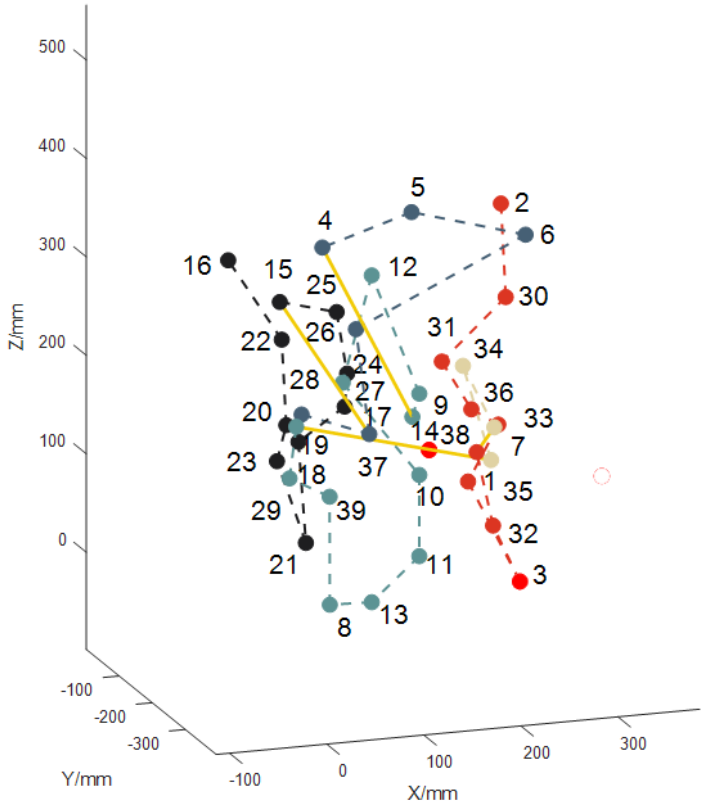
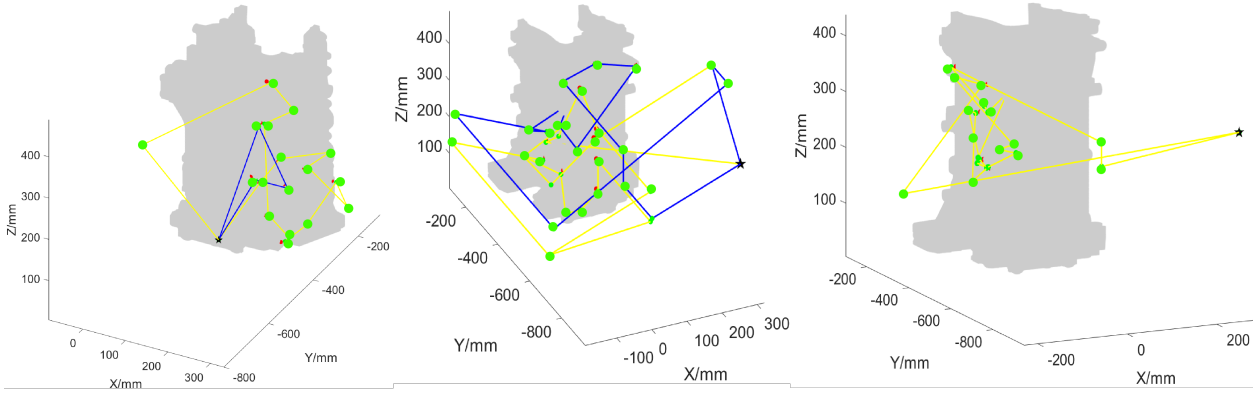


Figure 12. Inter-group and intra-group inspection sequence diagram, different color dots represent the MPs within different groups, the dotted line denotes the intra-group inspection sequence, the solid yellow line denotes the inter-group inspection sequence.



(a) **(b)** **(c)**

Figure 13. Inspection path diagram based on DSV algorithm. (**a**) Collision-free path for groups 1–2, in which yellow line represents the path of $\mathbf{S}_1$ and blue line represents the path of $\mathbf{S}_2$. (**b**) Collision-free path for groups 3–4, in which yellow line represents the path of $\mathbf{S}_3$ and blue line represents the path of $\mathbf{S}_4$. (**c**) Collision-free path for group 5.

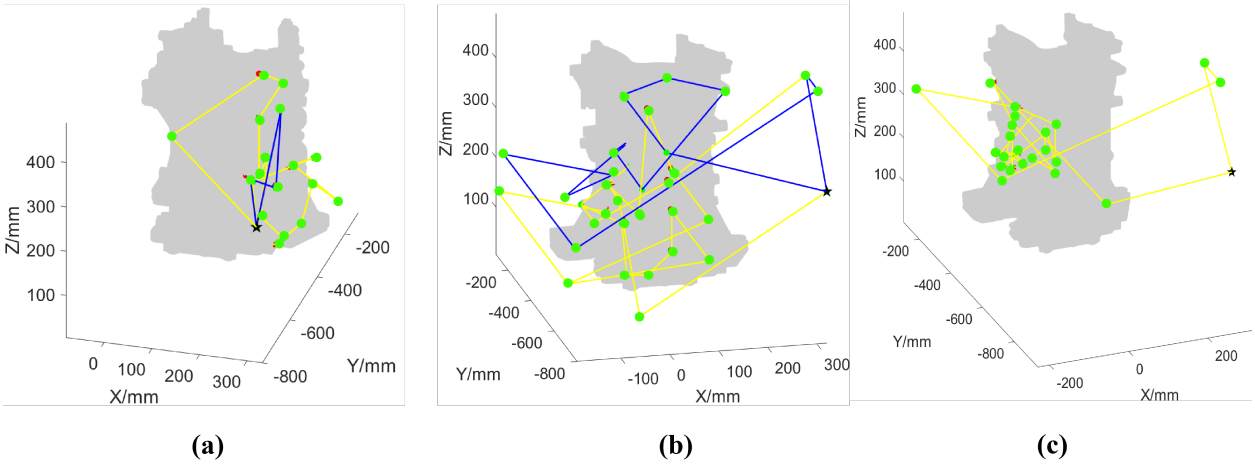


Figure 14. Inspection path diagram based on MSS algorithm. (**a**) Collision-free path for groups 1–2, in which yellow line represents the path of $\mathbf{S}_1$ and blue line represents the path of $\mathbf{S}_2$. (**b**) Collision-free path for groups 3–4, in which yellow line represents the path of $\mathbf{S}_3$ and blue line represents the path of $\mathbf{S}_4$. (**c**) Collision-free path for group 5.

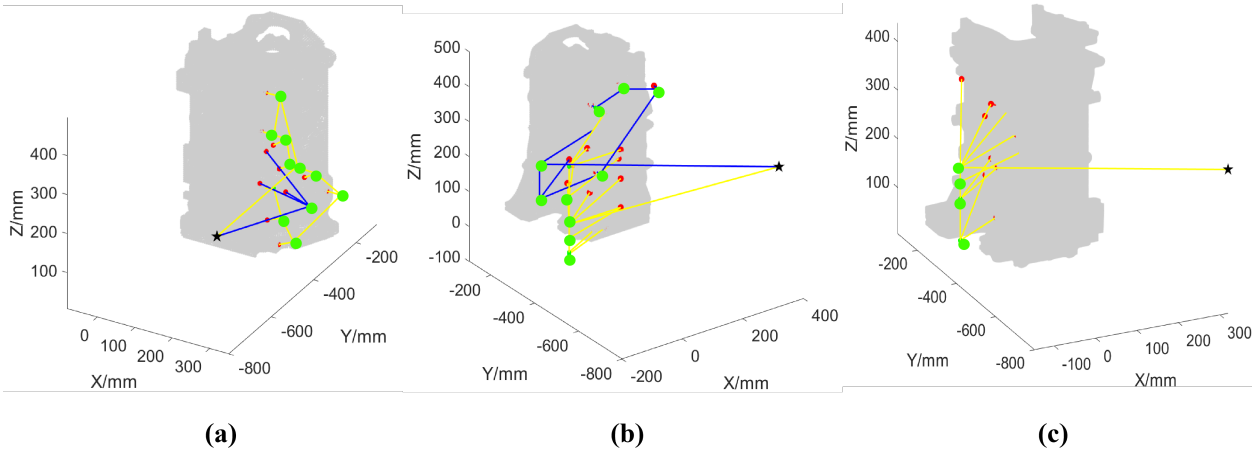


Figure 15. Inspection path diagram based on RRT-MRNC algorithm. (**a**) Collision-free path for groups 1–2, in which yellow line represents the path of $\mathbf{S}_1$ and blue line represents the path of $\mathbf{S}_2$. (**b**) Collision-free path for groups 3–4, in which yellow line represents the path of $\mathbf{S}_3$ and blue line represents the path of $\mathbf{S}_4$. (**c**) Collision-free path for groups 5.

4.2. Path Reusability Analysis

The path between MPs $\mathbf{Q}_{3,27}$ and $\mathbf{Q}_{3,13}$ in $\mathbf{S}_3$ is chosen to evaluate the path reusability of different algorithms. In order to better compare the adaptability of different methods for dynamic tasks, the following two strategies are introduced in this paper for comparison: (1) reuse of initial paths, that is, only the original path can be deleted with performing path reuse rather than re-planning the path. Figure 16 represent the reused inspection path diagrams based on DSV, MSS, and RRT-MRNC for the MPs in $\mathbf{S}_3$, respectively. (2) re-planning of dynamic tasks. Figure 17 represent the re-planning inspection path diagrams based on DSV, MSS for the MPs in $\mathbf{S}_3$, respectively. The comparison of planning results under different methods is given in Table 4. The path reuse method does not require additional calculation and teaching for planning dynamic tasks. Thus, the strategy of path reuse improves planning efficiency greatly. The total path generation time based on the proposed RRT-MRNC method are reduced by 55.2%, 54.9%, respectively. Furthermore, as can be seen from the Figure 16, when path reuse is conducted based on DSV and MSS algorithms, dynamic tasks only can be planned along the original path, and adding many redundant path. However, when path reuse is conducted based on the proposed

algorithm, the reusability of obstacle avoidance points within $\mathbf{x}_{init}$ ensures a collision-free path between any MPs, and the redundant path generated by MPs that currently do not need to be measured in the original path are deleted. Based on the above process, the inspection time is reduced and the efficiency of re-planning inspection is improved.

Table 4. Comparison of dynamic task planning results by different methods.

	Computation Time (s)	Motion Time (s)	Teaching Time (s)	Total Time (s)
Path reuse based on DSV	0.0	86.0	0.0	86.0
Path reuse based on MSS	0.0	85.4	0.0	85.4
Path re-planning based on DSV	337.5	61.1	129.8	528.3
Path re-planning based on MSS	277.5	69.4	147.5	494.5
Path reuse based on RRT-MRNC	0.0	38.5	0.0	38.5

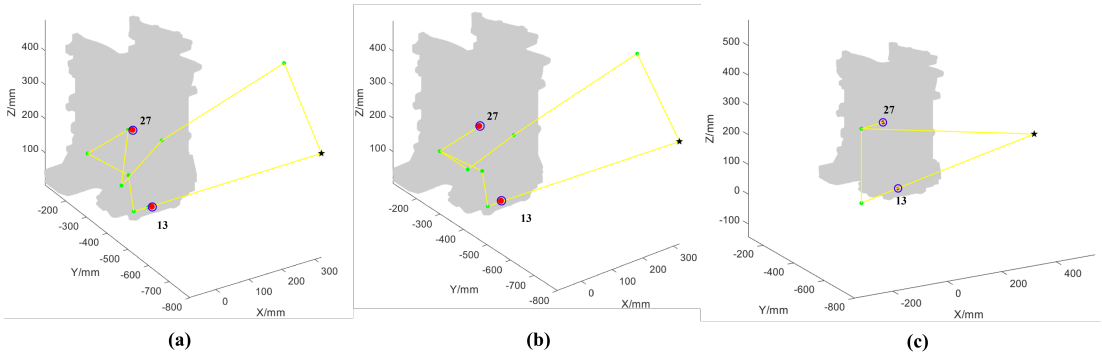


Figure 16. Schematic diagram of reuse path by different methods. (**a**) Reused path diagram based on DSV. (**b**) Reused path diagram based on MSS. (**c**) Reused path diagram based on RRT-MRNC.

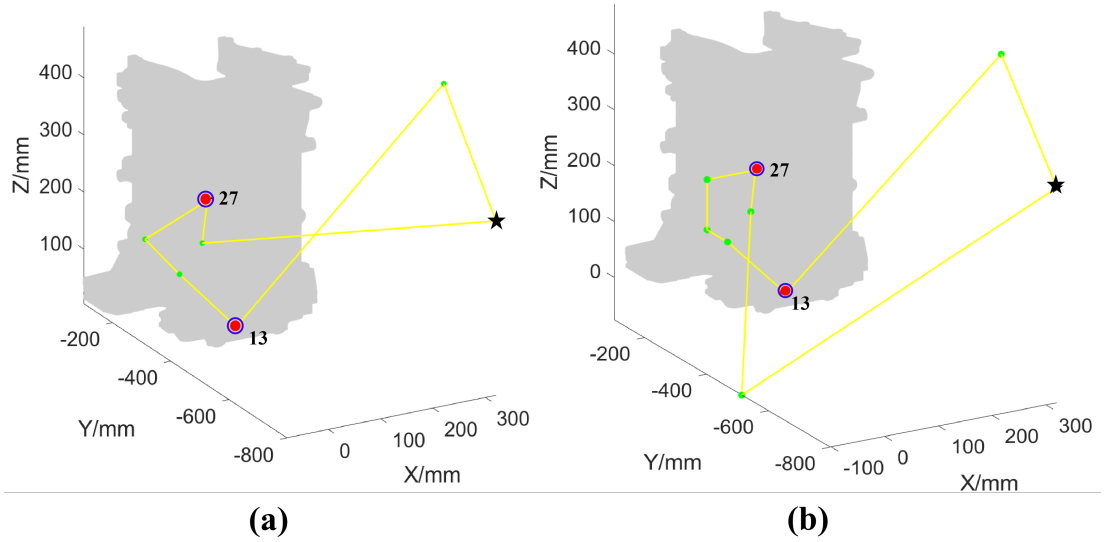


Figure 17. Schematic diagram of re-planning path by different methods. (**a**) Re-planning path diagram based on DSV. (**b**) Re-planning path diagram based on MSS.

4.3. Numerical Comparison

To improve the credibility of the experiments, the five groups of different MPs are randomly selected for path reusability experiments in the Table 2. Furthermore, the proposed RRT-MRNC method is compared with the state-of-the-art algorithms to verify the

effectiveness and feasibility of the proposed method, the results (motion time and total time) of the 5 simulations are further shown in Figure 18, where PRB DSV is the abbreviation of the path reuse based on DSV; PRB MSS is the abbreviation of path reuse based on MSS; PPB DSV is the abbreviation of path re-planning based on DSV; PPB MSS is the abbreviation of path re-planning based on MSS; PRB RRT-MRNC is the abbreviation of path reuse based on RRT-MRNC. The following summary can be given: (1) The dynamic task planning strategy based on path reuse takes less time compared to the dynamic task planning strategy based on path re-planning because the computation time and the teaching time are eliminated. (2) Compared with the benchmark methods, the proposed RRT-MRNC path reuse method effectively reduces the dynamic task planning time.

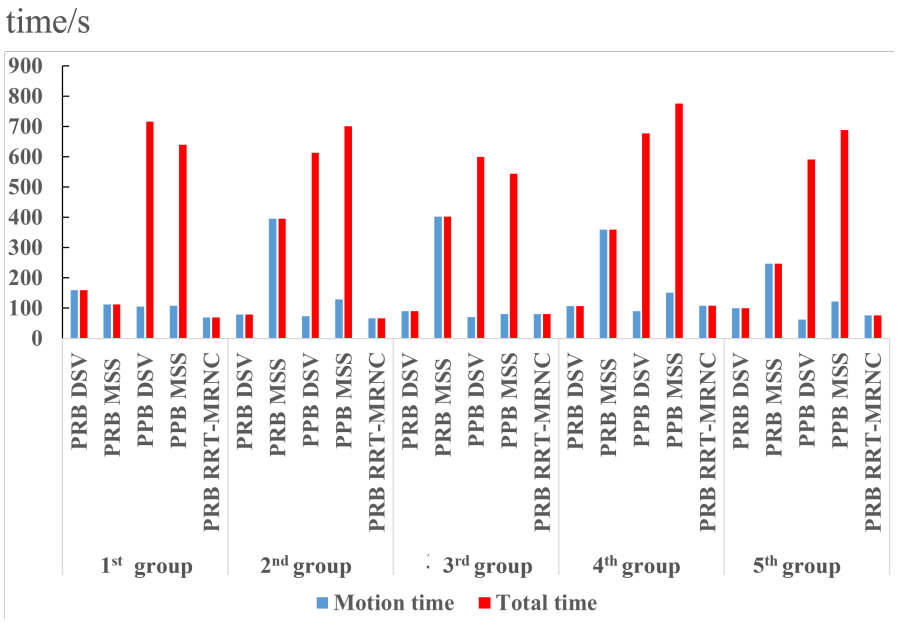


Figure 18. Comparison of 5 path planning results based on different methods.

5. Conclusions

A five-axis inspection path planning method considering inspection path planning efficiency, path length and probe rotation angle is proposed to improve the path reuse. Firstly, the measurement features are classified based on feasible inspection direction cone and accessibility of the MPs to ensure the minimum times of probe rotation. Secondly, the RRT-MRNC algorithm is proposed to implement local path planning considering inspection path reuse. Then, intra-group and inter-group planning paths are generated simultaneously based on the enhanced GA algorithm. The effectiveness of the proposed method is validated by using the simulation of the cylinder block inspection case. Results show that the re-planning time based on the RRT-MRNC method is shorter than the benchmark methods. Overall, the proposed method improves the adaptability of the algorithm to re-inspection, and improve the computer-aided inspection planning efficiency for the machining parts.

Author Contributions: Conceptualization, L.Y.; methodology, Z.W.; software, W.X.; writing—original draft preparation, Z.W. and W.X.; writing—review and editing, Z.W., W.X. and L.Y.; supervision, L.Y.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by National Natural Science Foundation of China (51875362) and Natural Science Foundation of Shanghai (21ZR1444500).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Anagnostakis, D.; Ritchie, J.;Lim, T.; Sivanathan, A.; Dewar, R.; Sung, R.; Bosché, F.; Carozza, L. Knowledge capture in CMM measurement planning: barriers and challenges. *Procedia CIRP* **2016**, *52*, 216–221.
2. Li, W.L.; Wang, G.; Zhang, G.; Li, Q.D.; Yin, Z.P.; Tang, K; Xie, C. Interference-free measurement path generation for impeller blades using an on-machine probe. *IEEE/ASME Transactions on Mechatronics* **2017**, *22*, 1218–1226.
3. Liu, Y.H.; Zhao, W.Z.; Sun, R.; Yue, X.W. Optimal path planning for automated dimensional measurement of free-form surfaces. *J. Manuf. Syst.* **2020**, *56*, 84–92.
4. Stojadinovic, S.M.; Majstorovic, V.D.; Gąska, A.; Sładek, J.; Durakbasa, N.M. Development of a coordinate measuring machine—based inspection planning system for industry 4.0. *Appl. Sci.* **2021**, *11*, 8411.
5. Slavenko M.S.; Vidosav D.M.; Numan M.D.; Tatjana V.S. Ants colony optimisation of a measuring path of prismatic parts on a CMM. *Metrol. Meas. Syst.* **2016**, *23*, 119–132.
6. Liu, Y,H.; An, C.; Duan, Z.X. A Safety-Space-Based approach to inspection path planning for the sheet metal assemblies. In Proceedings of the ASME 2021 International Mechanical Engineering Congress and Exposition, Virtual, USA, 1–5 November 2021.
7. Ren, M.J.;Kong, L.B.; Sun, L.J.; Cheung, C.F. A curve network sampling strategy for measurement of freeform surfaces on Coordinate Measuring Machines. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 3032–3043.
8. Chen, X.; You, X.; Jiang, J.; Ye, J.; Wu, H. Trajectory planning of dual-Robot cooperative assembly. *Machines* **2022**, *10*, 689.
9. Bai, Y.; Ding, X.; Hu, D.; Jiang, Y. Research on dynamic path planning of multi-AGVs based on reinforcement learning. *Appl. Sci.* **2022**, *12*, 8166.
10. Han, Z.H.; Liu, S.G.; Yu, F.; Zhang, X.D.;Zhang, G.X. A 3D measuring path planning strategy for intelligent CMMs based on an improved ant colony algorithm. *Int. J. Adv. Manuf. Technol.* **2017**, *93*, 1487–1497.
11. Li, B.; Feng, P.F.; Zeng, L.; Xu, C.; Zhang, J.F. Path planning method for on-machine measurement of aerospace structures based on adjacent feature graph. *Robot. Comput.-Integr. Manuf.* **2018**, *54*, 17–34.
12. Zhou, Z.;Zhang, Y;Tang, K. Sweep scan path planning for efficient free-form surface measurement on five-axis CMM. *Comput.-Aided Des.* **2016**, *77*, 1–17.
13. Devaurs, D.; Siméon, T.; Cortés, J. Optimal path planning in complex cost spaces with sampling-based algorithms. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 415–424.
14. Slavenko, M. S; Vidosav, D. M.; Numan, M. D.; Tatjana V. S. Towards an intelligent approach for CMM measurement planning of prismatic parts. *Measurement* **2016**, *92*, 326–339.
15. Zhao, H.B.; Jean, P.K.;Nick, V. G.; Bart, B.; Philip, B. Automated dimensional measurement planning using the combination of laser scanner and tactile probe. *Measurement* **2012**, *45*, 1–17.
16. Suh, J.; Gong, J.; Songhwai, O. Fast sampling-based cost-aware path planning With nonmyopic extensions using cross entropy. *IEEE Trans. Robot.* **2017**, *13*, 1313–1326.
17. Lai, T.; Morere, P.; Ramos, F.; Francis, G. Bayesian local sampling-based planning. *IEEE Trans. Ind. Electron.* **2020**, *5*, 1954–1961.
18. Qi, J.; Yang, H.; Sun, H.X. MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment. *IEEE Robot. Autom. Lett.* **2021**, *68*, 7244–7251.
19. Yang, L.; Fu, L.; Li, P.; Mao, J.; Guo, N. An effective dynamic Path Planning Approach for Mobile Robots Based on Ant Colony Fusion Dynamic Windows. *Machines* **2022**, *10*, 7244–7251.
20. Li, D.C.; Yin, W.P.; Wong, W.E.J.; Chau, M.Y.M. Quality-oriented hybrid path planning based on A* and Q-Learning for unmanned aerial vehicle. *IEEE Access* **2022**, *10*, 7664–7674.
21. Wang, B.; Liu, Z.; Li, Q.B.; Prorok, A. Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning. *IEEE Robotics and Automation Letters* **2020**, *5*, 6932–6939.
22. Li, X.; Liu, H.; Dong, M; Prorok, A. A General Framework of Motion Planning for Redundant Robot Manipulator Based on Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics* **2021**, *18*, 5253–5263.
23. Mussa, M.; David, J.; Michaël, R.; Ahmed, I.; Jean, F. 3D part measurement path planning of a laser scanner with control on the uncertainty. *Computer-Aided Design* **2011**, *43(4)*, 345–355.
24. Yi, B.W.; Qiao, F.; Li, H.; Wang, X.S.; Wu, S.J.; Huang, N. Touch trigger probe-based interference-free measurement path planning for free-form surfaces by optimizing the probe posture. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–8.
25. Zhang, S.G.;Ajmal, A.;Wootton, J.;Chisholm, A. A feature-based measurement process planning system for co-ordinate measuring machine (CMM). *Journal of Materials Processing Technology* **2000**, *107*, 111–118.
26. Li, Y.M.; Zeng, L.; Tang, K.; Xie, C. Orientation-point relation based measurement path planning method for 5-axis OMI system. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101827.
27. Anagnostakis, D.; Ritchie, J.; Lim, T.; Sung, R.; Dewar, R. A Virtual CMM measurement Tool for Capturing Planning Strategies. In Proceedings of the ASME 2017 International Mechanical Engineering Congress and Exposition,Cleveland, Ohio, USA. 6–9 August 2017.
28. Martínez-Pelliter, S.; Barreiro, J.; Cuesta, E.; Fernández-Abia, A.I. Knowledge base model for automatic probe orientation and configuration planning with CMMs. *Robot. Comput.-Integr. Manuf.* **2018**, *49*, 285–300.

29. Wang, S.Q. Research on randomized greedy algorithm for k-median problem. In Proceedings of the 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering, Bali, Indonesia, 04-07 August 2011; 98–101.
30. Liu, Y.H.; Zhao, W.Z.; Liu, H.P.; Wang, Y.N.; Yue, X.W. Coverage Path Planning for Robotic Quality Inspection With Control on Measurement Uncertainty. *IEEE/ASME Transactions on Mechatronics* **2022**, doi: 10.1109/TMECH.2022.3142756.
31. Chen, L.; Yang, H.; Shan, Y.X.; Tian, W.; Li, B.J.; Cao, D.P. A Fast and Efficient Double-Tree RRT*-Like Sampling-Based Planner Applying on Mobile Robotic Systems. *IEEE/ASME Transactions on Mechatronics* **2018**, *23*, 2568–2578.
32. Gohar, V.; Mehdi, Y.; Mahdieh, P.; Naghibi, S. A New Approach to Solve Traveling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator. In Proceedings of the International Conference of Soft Computing and Pattern Recognition, Malacca, Malaysia, 4-7 December 2009; 112–116.
33. Li, Z.; Janardhanan, M.N.; Ponnambalam, S.G. Cost-oriented robotic assembly line balancing problem with setup times: Multi-objective algorithms. *J. Intell. Manuf.* **2021**, *32*, 989–1007.