Rotation Posture Optimization of Large Spacecraft Cabin During Robotic In-Situ Machining

LIU Shaorui, TIAN Wei, SHEN Jianxin, LI Bo*

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, P.R. China

(Received 5 September 2022; revised 25 December 2022; accepted 20 June 2023)

Abstract: When robotic systems are used to perform the in-situ machining of the large spacecraft cabin, there is a dual-objective optimization issue concerning the numbers of cabin's rotation and the overall machining performance of robots. An optimization method of cabins rotation scheme is proposed based on the robots' stiffness characteristic and the non-dominated sorting genetic algorithm (NSGA)-Ⅱ. First, a quality evaluation index for all machining features is proposed based on the Cartesian stiffness of robots, and the process of machining is analyzed and modeled. Second, to utilize the NSGA-Ⅱ, a double chromosome coding method is proposed to encode the machining process, corresponding crossover and mutation operator is also designed. Third, to solve the frequent appearing of illegal codes, a repair operator that ensures the population's evolutionary efficiency is specially designed based on the problem structure. Finally, the result of a case study shows that increasing the number of rotation postures to a certain extent can effectively improve the robots' machining performance, and achieve a comprehensive optimization of the mission's time efficiency and machining quality.

Key words: large structure manufacturing; in-situ machining; robot stiffness optimization; dual-objective optimization; non-dominated sorting genetic algorithm (NSGA)-Ⅱ

CLC number: TP242.2 **Document code:** A **Article ID:** 1005-1120(2023)03-0239-14

0 Introduction

Large structures like aircraft fuselage or spacecraft cabin are commonly seen in the field of aerospace. The machining of these high-value, large-dimension products with complex structures directly affect the productivity and cost of the overall manufacturing process. In a conventional workshop, products are transported toward large dedicated machine tools by crane or transporter systems to complete the machining. However, due to the dimension and weight of the large structures, the transportation could be time-consuming and dangerous. For this reason, in-situ machining which keeps the product installed on one clamp system and moves the in-situ machine tools toward the product, can offer obvious advantages in large structure machining[1]. An ideal in-situ machine tool shall be light-weight, flexible and compact for easy deployment in the workshop. Recently developed robotic machining systems have these exact qualities[2-3]. These are high-accuracy industrial robots equipped with mobile platforms and visual servo or secondary feedback systems[4] that can move and operate freely on the shop floor, thus they are more capable of handling various large structures than large dedicated machine tools[5-6].

This paper considers the manufacturing mission of a large spacecraft cabin where the discrete machining tasks are handled by multiple robots. Due to the structural characteristic of the cabin, several rotations are required during the machining process. Since the scheme of rotation has an influence to the overall performance of the system, the optimal scheme has to be specified for the mission.

*Corresponding author, E-mail address: tw_nj@nuaa.edu.cn.

How to cite this article: LIU Shaorui, TIAN Wei, SHEN Jianxin, et al. Rotation posture optimization of large spacecraft cabin during robotic in-situ machining[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2023, 40(3): 239-252.

http://dx.doi.org/10.16356/j.1005-1120.2023.03.001

The Cartesian stiffness of an industrial robot is significantly lower than typical computer numerical control (CNC) machines[7], which causes the major obstacle of robotic machining: The unsatisfactory machining quality. Extensive studies have been done on the stiffness characteristics of industrial robots[8-9], and it has been made clear that the Cartesian stiffness of a robot is highly correlated with its configuration[10-11]. Therefore, prior to investing in new robots with high structural stiffness[12], optimizing the robot configuration for machining should be first considered in the task planning of a robotic machining system.

The inherent redundant degree of freedom (DOF) of industrial robots in five-axis machining, as well as the extra DOF provided by the mobile platforms, is essential to the optimization of robot configuration. Bu et al. studied the stiffness optimization of a robotic drilling system that was used in aircraft wing structure assembly[13-14], where the redundant DOF with respect to the tool axis was utilized for configuration optimization, and the experimental results showed significant quality improvements. Guo et al. verified the correlation between the proposed stiffness index and the translational displacement of end-effector caused by external forces[15]. The rivet hole quality was then improved through the configuration optimization of a robotic drilling system. Jiao et al. utilized the external axis of a robotic drilling system to optimize the tool axial stiffness at drilling process, and improved the hole roundness and consistency[16]. Aiming at the stiffness optimization of robotic milling, Xiong et al. proposed a configuration optimization method that converted the CNC tool path into the robot trajectory[17]. The redundant DOF was utilized by a configuration optimization model. For tasks or products more complex than single point processing (e.g., drilling), some performance indices that are more comprehensive than axial stiffness are needed[18-19]. Fan et al. proposed the mean stiffness performance index (MSPI) for the robotic grinding mission of a large turbine blade[19], where MSPI provided a comprehensive performance evaluation to a large grinding region, and the robot base position as well as the configuration were optimized according to MSPI.

In in-situ machining, a multi-robot team is favored for its advantage in production efficiency. Some widely used multi-robot work cells would consider the position optimization of workpieces, since it strongly affects the feasibility and performance of the operation. Mutti et al. proposed an algorithm for this purpose[20], where the operation feasibility, including kinematic feasibility (e.g., joint limit) and collision avoidance, were considered and optimized. There are similar studies that aim to improve operation feasibility or productivity of multi-robot work cells, but the robot base positions are optimized instead of the position of workpiece[21-22]. However, seldom do multi-robot work cells perform machining tasks. These studies of position optimization do not take the stiffness or machining performance into consideration. Moreover, the stiffness evaluation of a team of robots handling complex machining tasks lacks discussion in previous studies.

The existing studies on stiffness/configuration optimization focus on the application scenario of a single robot performing one specific task, rather than a team of robots performing a group of tasks. Moreover, the stiffness index is often the single objective considered, or only combined with kinematic performance[16-19]. However, optimizing the machining performance may affect other performances that the manufacturers want to improve, e.g., the production efficiency. In such case, it may be inappropriate to take only robot performance into consideration.

This paper studies the machining solution optimization of a large spacecraft cabin, where the rotation of the cylindrical cabin (controlled by the positioner) provides a mean to optimize the robot configuration and the stiffness in machining. However, the rotation is expected to be as less as possible, so that the overall time cost can be reduced and maintain productivity. We therefore define the problem as a dual-objective optimization problem, and design an algorithm based on the non-dominated sorting genectic algorithm (NSGA)-Ⅱ to provide Pareto-optimal solutions[23-24]. A quality evaluation index of the whole cabin is proposed to guide the algorithm's optimization process.

The rest of the paper is organized as follows: Section 1 introduces the real-world motivation of the study. Section 2 describes the dual-objective optimization problem in detail and proposes a quality evaluation index of the whole cabin. Section 3 introduces the stiffness index calculation of the robot. Section 4 designs the algorithm. Section 5 conducts a case study of the machining solution optimization of a cabin and discusses the results. Section 6 concludes the study and discusses the object of future research.

1 Real-World Motivation

The machining task of the large spacecraft cabin's surface features is performed by high-precision robotic systems in the way of in-situ machining. As shown in Fig.1, the main part of the cabin is a cylindrical, thin-walled structure which is composed of several cylindrical segments, and there are many small supports distributed on its surface. These supports are used as mounting points for the exterior equipment (i.e., sensors, radar antenna, and space manipulator). On top of the supports are the mounting surfaces that require the robots to process. Since the equipment have various shapes and are mounted in different ways, the supports are distributed non-uniformly and have mounting surfaces facing in various directions. All the mounting surfaces have a certain machining allowance left, so that the errors caused by deformation and assembly can be eliminated by in-situ measuring and machining, and eventually ensure the performance and precision of the cabin's exterior equipment.

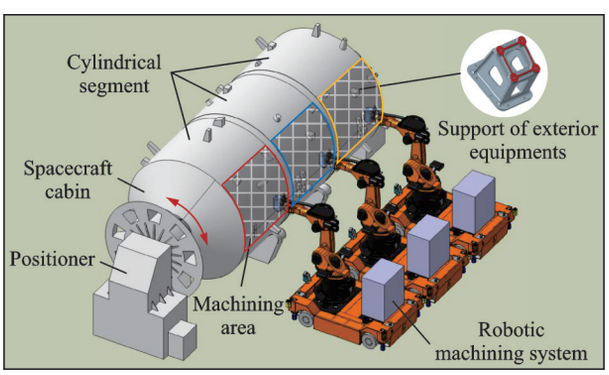

Fig.1 A scenario of large spacecraft cabin in-situ machining by robotic systems

The cabin is positioned horizontally, and each cylindrical segment is handled by one or two robots (on one side or both sides of the cabin). Due to the limitation of robot workspace, several rotations, i.e., repositioning will be needed to complete the whole cabin. The number and angle of rotations, which is the rotation scheme (RS), will directly change the position and posture of a support that appears in the robot's workspace, and therefore influence the robot's configuration as well as its stiffness in machining, and eventually influence the quality of the machining. In theory, increasing the number of rotations helps to improve the machining stiffness of more mounting surfaces. Yet, this will introduce more interruptions and more recalibration workload to the whole process, resulting in lower productivity and higher production cost. Based on the above reasons, the requirement of the RS optimization can be expressed as: Maximizing the machining quality of the cabin while reducing the number of rotations needed; or specifying the minimum number of rotations needed for a given quality index.

In Fig.1, the position optimization of robotic systems is not considered due to the following reasons: The robots all use relatively wide chasses to ensure stability, consequently. There is little room left on the shop floor for adjusting positions, hence the factor of robot position has less influence on the problem. Therefore, we treat them as fixed units. Moreover, the thin-walled cabin has tooling systems both inside and at the bottom, therefore it has negligible gravity deformation during the rotation and repositioning.

2 Robot Stiffness Index

A robot stiffness index is proposed as a mean of machining performance evaluation. Salisbury et al. proposed the widely used static stiffness model[25]

$$\boldsymbol{K}(\boldsymbol{\theta})=\boldsymbol{J}(\boldsymbol{\theta})^{-\mathrm{T}}\boldsymbol{K}_{\theta}\boldsymbol{J}(\boldsymbol{\theta})^{-1} \tag{1}$$

where $\boldsymbol{K}$ is the Cartesian stiffness matrix; $\boldsymbol{\theta}=[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^{\mathrm{T}}$ the robot configuration; $\boldsymbol{J}(\boldsymbol{\theta})$ the Jacobian matrix; $\boldsymbol{K}_{\theta}=\mathrm{diag}(k_1, k_2, k_3, k_4, k_5, k_6)$ the joint stiffness matrix. In this model, and the Cartesian stiffness is only concerned with joint stiffness and robot configuration. $\boldsymbol{K}(\boldsymbol{\theta})$ consists of four sub-matrices

$$\boldsymbol{K}(\boldsymbol{\theta})=\begin{bmatrix}\boldsymbol{K}_{fd}(\boldsymbol{\theta}) & \boldsymbol{K}_{f\delta}(\boldsymbol{\theta})\\ \boldsymbol{K}_{md}(\boldsymbol{\theta}) & \boldsymbol{K}_{m\delta}(\boldsymbol{\theta})\end{bmatrix} \tag{2}$$

where $\boldsymbol{K}_{fd}$ and $\boldsymbol{K}_{f\delta}$ are the force-translation and force-rotation matrices; $\boldsymbol{K}_{md}$ and $\boldsymbol{K}_{m\delta}$ the torque-translation and torque-rotation matrices. Previous studies have demonstrated that the linear displacement of the end-effector (EE) caused by cutting force is the main influencing factor of the machining quality[20-21], while the torque has negligible influence. Therefore, the displacement model of EE can be simplified as

$$\boldsymbol{f}=\boldsymbol{K}_{fd}(\boldsymbol{\theta})\boldsymbol{d} \tag{3}$$

where $\boldsymbol{f}=[f_x, f_y, f_z]$ is the force applied on EE; and $\boldsymbol{d}=[d_x, d_y, d_z]$ the linear displacement of EE. The unit of $\boldsymbol{f}$, $\boldsymbol{d}$ and $\boldsymbol{K}_{fd}(\boldsymbol{\theta})$ are N, mm and N/mm, respectively. Considering a unit force $\boldsymbol{f}_u$, there is

$$\|\boldsymbol{f}_u\|^2=\boldsymbol{f}_u^{\mathrm{T}}\boldsymbol{f}_u=\boldsymbol{d}_u^{\mathrm{T}}\boldsymbol{K}_{fd}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{K}_{fd}(\boldsymbol{\theta})\boldsymbol{d}_u=1 \tag{4}$$

where $\boldsymbol{d}_u$ is the displacement of EE when $\boldsymbol{f}_u$ is applied.

Eq.(4) defines the stiffness ellipsoid which changes in shape and volume with the robot configuration $\boldsymbol{\theta}$. The length of the ellipsoid's semi-axis in the direction of which cutting force is applied reflects the robot stiffness in the machining, where the stiffness along the tool feed force and radial force directions affects the trajectory accuracy, and the stiffness along the tool axial force direction affects the quality of the machined surface.

Typically, when setting the tool frame of EE, one axis of the frame shall coincide with the tool axis, while the tool feed direction coincides with (or opposite to) another axis of the frame, as shown in Fig.2. Consequently, the tool axial force, the feed force and the radial force act in the directions collinear with the three tool frame axes. The length of the three semi-axes along the tool frame's x, y, z directions are denoted as λ_{tx}, λ_{ty}, and λ_{tz}, respectively. These values are the square of the stiffness index, thus the robot stiffness index in machining can be defined as

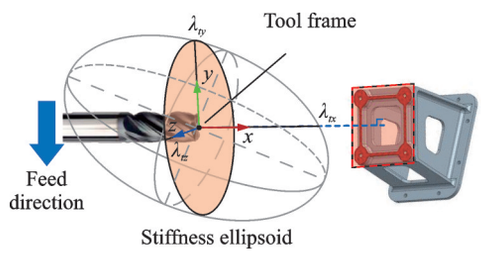

Fig.2 Stiffness ellipsoid and its semi-axes

$$k=\sqrt[6]{\lambda_{tx}\lambda_{ty}\lambda_{tz}} \tag{5}$$

The dimension of k is N/mm. k is concerned with $\boldsymbol{\theta}$, while $\boldsymbol{\theta}$ is determined by the machining task, that is, the support to the machine, where the robot is positioned, and what rotation angle the cabin currently stops at. Thus in a given machining solution where $\boldsymbol{\theta}$ is determined for each machining task, the stiffness index k of any task is a determined value. It is then used as a machining quality evaluation for the corresponding task.

3 Problem Formulation

Let $\boldsymbol{T}=\{t_1, t_2, \cdots, t_n\}$ denote the n mounting surfaces that need to be machined, which are the tasks. The number of rotation m, the angles at which the cabin positioned after each rotation, $\boldsymbol{R}=[r_1, r_2, \cdots, r_m]$, and the tasks to be machined at each angle, $\boldsymbol{A}=\{\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_m\}$, form the machining solution of the cabin. A feasible solution is defined as $s=<\boldsymbol{R}, \boldsymbol{A}>$. Then there are constraints

$$\begin{cases} C1{:}\boldsymbol{A}_i\subset\boldsymbol{T}^{\mathrm{rch}}(r_i) & i=1,2,\cdots,m\\ C2{:}\boldsymbol{A}_i\neq\varnothing & i=1,2,\cdots,m\\ C3{:}\boldsymbol{A}_i\cap\boldsymbol{A}_j=\varnothing & i\neq j\\ C4{:}\bigcup_{i=1}^{m}\boldsymbol{A}_i=\boldsymbol{T} \end{cases} \tag{6}$$

where $C1$ means the tasks to be machined at each angle have to be within the robots' workspace: $\boldsymbol{T}^{\mathrm{rch}}(r_i)$ are the task set located within robot workspace when the cabin is positioned at angle r_i; $C2$ means there are no redundant rotations (no tasks assigned); $C3$ and $C4$ mean any task shall be processed at one and only one angle. To ease modeling and calculation, the rotation angle will be discretized in a small step θ_{step} (i.e. $\theta_{\mathrm{step}}=10^\circ$), and $r_i=\theta_{\mathrm{step}}\cdot q$ ($i=1, 2, \cdots, m$; q is a integer).

When processing at different angles, the position and posture of a support and its mounting surface will be significantly different, and the robot will have to reach it in a different way, thus lead to very different robot configurations and stiffness indices. Generally, each surface can only achieve high pro-

cessing stiffness at very limited angles, to improve the machining quality of the whole cabin, more rotations will be needed. However, in practice, we hope the rotation number m to be a value as small as possible, so the first objective function f_1 is given as

$$\min f_1 = m \tag{7}$$

In a given solution $\dot{s}$, the configuration as well as the stiffness index of the robot who processes task t_j ($t_j \in T$) totally depends on $\dot{s}$, therefore we denote the machining stiffness of t_j as $k_j(\dot{s})$. Since the machining quality is directly correlated to the stiffness, $k_j(\dot{s})$ can be used as the quality evaluation index of t_j. On this basis, the quality evaluation of the whole cabin should be a function of the machining stiffness of all tasks. $\boldsymbol{K}(\dot{s}) = \{k_1(\dot{s}), k_2(\dot{s}), \cdots, k_n(\dot{s})\}$ can be denoted as $F(\boldsymbol{K}(\dot{s}))$. The objective function of the whole cabin's machining quality f_2 is given as

$$\max f_2 = F\left(\boldsymbol{K}(\dot{s})\right) \tag{8}$$

It is obvious that f_1 and f_2 are two conflicting goals: A solution s shall not likely achieve optimum quality goals with minimum number of rotation angles, and only Pareto optimal solutions will exist for the problem.

As mentioned above, the stiffness index is used to evaluate the robot performance in a certain machining task. For a given machining solution $\dot{s}$ to the cabin with n supports, there will be n stiffness indices ($\boldsymbol{K}(\dot{s}) = \{k_1(\dot{s}), k_2(\dot{s}), \cdots, k_n(\dot{s})\}$). It is essential to provide a stiffness-based global performance evaluation index for solution $\dot{s}$, which is the $F\left(\boldsymbol{K}(\dot{s})\right)$.

The global index should properly reflect the n values in $\boldsymbol{K}(\dot{s})$. The mean value of $\boldsymbol{K}(\dot{s})$, denoted by $\bar{k}(\dot{s})$, might be a choice of the global index. However, given the fact that undesirable results are more likely to occur to tasks with lower stiffness values, evaluating the global performance of $\dot{s}$ should primarily focus on the lower values in $\boldsymbol{K}(\dot{s})$, yet $\bar{k}(\dot{s})$ could easily cover up the lower values.

To provide a more reasonable global performance index than $\bar{k}(\dot{s})$, we propose two principles for designing the index:

(1) The increase of any k in $\boldsymbol{K}(\dot{s})$ should let the global index $F\left(\boldsymbol{K}(\dot{s})\right)$ increase.

(2) $F\left(\boldsymbol{K}(\dot{s})\right)$ should be more sensitive to the changes of the lower values in $\boldsymbol{K}(\dot{s})$ than the higher values, which means improving a lower stiffness value should contribute more to $F\left(\boldsymbol{K}(\dot{s})\right)$ than improving a value that is already high.

Here it is defined that the median value of $\boldsymbol{K}(\dot{s})$ is the dividing line between the lower and the higher values in $\boldsymbol{K}(\dot{s})$. To fulfill the two principles, a weighted mean stiffness index (WMSI) is proposed as $F\left(\boldsymbol{K}(\dot{s})\right)$. The median, the upper and the lower quartiles of $\boldsymbol{K}(\dot{s})$ are denoted as k_{med}, k_{Q3}, and k_{Q1}, respectively, and then $\boldsymbol{K}(\dot{s})$ is divided into four subsets

$$\begin{cases} \boldsymbol{K}_{h_1} = \{k_x \mid k_x \in \boldsymbol{K}(\dot{s}) \;\; k_{\text{med}} < k_x \leqslant k_{\text{Q3}}\} \\ \boldsymbol{K}_{h_2} = \{k_x \mid k_x \in \boldsymbol{K}(\dot{s}) \;\; k_x > k_{\text{Q3}}\} \\ \boldsymbol{K}_{l_1} = \{k_x \mid k_x \in \boldsymbol{K}(\dot{s}) \;\; k_{\text{Q1}} < k_x \leqslant k_{\text{med}}\} \\ \boldsymbol{K}_{l_2} = \{k_x \mid k_x \in \boldsymbol{K}(\dot{s}) \;\; k_x \leqslant k_{\text{Q1}}\} \end{cases}$$

A boxplot is used to demonstrate the division in Fig.3. Both the higher and lower values are divided into two levels, which are $\boldsymbol{K}_{h_2}$, $\boldsymbol{K}_{h_1}$ and $\boldsymbol{K}_{l_1}$, $\boldsymbol{K}_{l_2}$. Different weighting factors shall be applied to the four subsets when calculating WMSI, so that WMSI will have different sensitivity to the values in different subsets.

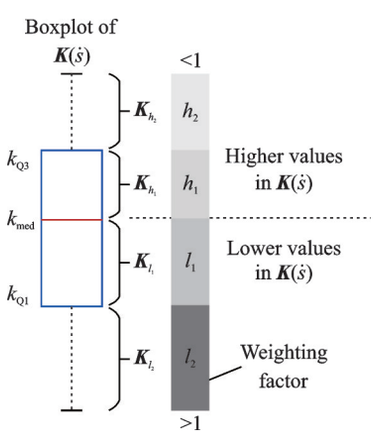

Fig.3 Four subsets of $\boldsymbol{K}(\dot{s})$

The global performance index, i.e. WMSI, is defined in Eq.(9), where h_2, h_1, l_1, l_2 are the weighting factors of $\boldsymbol{K}_{h_2}$, $\boldsymbol{K}_{h_1}$, $\boldsymbol{K}_{l_1}$, and $\boldsymbol{K}_{l_2}$, and there are $l_2 > l_1 > 1 > h_1 > h_2 > 0$, and $h_1 + l_1 = h_2 + l_2 = 2$. Since $\boldsymbol{K}_{l_2}$ and $\boldsymbol{K}_{l_1}$ have more weights than $\boldsymbol{K}_{h_2}$ and $\boldsymbol{K}_{h_1}$, WMSI will be more sensi-

tive to the values in $\boldsymbol{K}_{l_2}$ and $\boldsymbol{K}_{l_1}$. And when there are $\boldsymbol{K}(\dot{s})$ and $\boldsymbol{K}(\ddot{s})$ that have equal mean values $(\bar{k}(\dot{s})=\bar{k}(\ddot{s}))$, if $\sum \boldsymbol{K}_{l_2}(\dot{s})>\sum \boldsymbol{K}_{l_2}(\ddot{s})$ and $\sum \boldsymbol{K}_{l_1}(\dot{s})>\sum \boldsymbol{K}_{l_1}(\ddot{s})$, then there must be $F\left(\boldsymbol{K}(\dot{s})\right)>F\left(\boldsymbol{K}(\ddot{s})\right)$. This means that solution $\dot{s}$ outperforms $\ddot{s}$, because the lower part of $\boldsymbol{K}(\dot{s})$ is overall higher than the lower part of $\boldsymbol{K}(\ddot{s})$.

$$\mathrm{WMSI}=F\left(\boldsymbol{K}(\dot{s})\right)=\left[h_2\sum \boldsymbol{K}_{h_2}+h_1\sum \boldsymbol{K}_{h_1}+l_1\sum \boldsymbol{K}_{l_1}+l_2\sum \boldsymbol{K}_{l_2}\right]\Big/n \tag{9}$$

4 Algorithm Design

The NSGA-Ⅱ algorithm is used to solve the dual-objective optimization problem proposed in Section 2. The main idea of genetic algorithm (including NSGA-Ⅱ) is to find near-optimal solutions by repeated propagation and selection of a given population, in which each individual is an encoded solution. A double chromosome encoding method is first proposed in Section 4.1, then its crossover, repair and mutation operators for propagation are designed in Section 4.2. The main loop of the algorithm is introduced in Section 4.3.

4.1 Double chromosome coding method

Encoding the solutions in an effective way is the precondition of using genetic algorithms. According to the definition of feasible machining solutions ($s=<\boldsymbol{R}, \boldsymbol{A}>$), we propose a double chromosome method to encode any instance of s. The first chromosome encodes the rotation scheme of the cabin, which is the $\boldsymbol{R}$ in s. The second chromosome encodes the allocation of the tasks, which is the $\boldsymbol{A}$ in s:

(1) $\boldsymbol{V}_R$ is a vector of length $m_{\_\mathrm{UB}}$, the ith element is the rotation angle r_i. $m_{\_\mathrm{UB}}$ is the maximum number of rotations allowed, when $m < m_{\_\mathrm{UB}}$ (the number of rotations). The empty slots in $\boldsymbol{V}_R$ are filled with Null.

(2) $\boldsymbol{V}_A$ is a vector of length n (the number of tasks), the jth element $r^{(j)}$ is the rotation angle that task t_j is assigned to.

$\boldsymbol{V}_R$ is referred to as the first chromosome, and $\boldsymbol{V}_A$ as the second chromosome. Solution s can then be expressed by the chromosomes as $<\boldsymbol{V}_R, \boldsymbol{V}_A>$ (Fig.4).

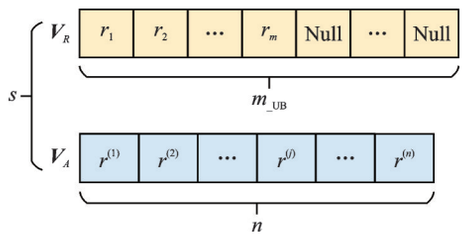

Fig.4 Double chromosome coding of solution s

As we propose the encoding method, it is also necessary to clarify the legality of given codes, i.e. whether $<\boldsymbol{V}_R, \boldsymbol{V}_A>$ describes a feasible solution that satisfies all constraints. The constrains $C1$—$C4$ in Eq.(1) are rewritten in the encoded form as

$$\begin{cases} C'1: \bigcup\limits_{r\in V_R} T^{\mathrm{rch}}(r)=T \\ C'2: \forall r^{(i)}\in \boldsymbol{V}_A \quad r^{(i)}\in \boldsymbol{V}_R \wedge t_i\in T^{\mathrm{rch}}\left(r^{(i)}\right) \\ C'3: \forall r\in \boldsymbol{V}_R \quad r\in \boldsymbol{V}_A \end{cases} \tag{10}$$

where $C'1$ means the rotation scheme proposed by $\boldsymbol{V}_R$ allows the robots to reach all tasks; $C'2$ means any rotation angle the tasks assign to exist in the rotation scheme $\boldsymbol{V}_R$, and the corresponding task can be processed under that angle; $C'3$ means any rotation angle in $\boldsymbol{V}_R$ that has been assigned tasks (exist in $\boldsymbol{V}_A$), which means there are no redundant rotations exist in the scheme.

The legality constraints in Eq.(10) are in effect throughout the operations concerning with codes, including the creation of initial population, the crossover of parents, and the mutation of new codes, and we refer to the codes that do not satisfy the constraints as illegal codes.

4.2 Operators design for propagation

4.2.1 Crossover operator

Crossover is the way to combine the genetic information of two parents. We use bi-tournament selections to select competitive parents s_{p1} and s_{p2} from the population, and then perform multi-point crossover to their second chromosome, $\boldsymbol{V}_A^{p1}$ and $\boldsymbol{V}_A^{p2}$, as shown Fig.5. The chromosomes produced by the crossover are denoted as $\boldsymbol{V}_A^{c1}$ and $\boldsymbol{V}_A^{c2}$. If the offspring inherit the first chromosome directly from their parents, the codes of the offspring are $s_{c1}=<\boldsymbol{V}_R^{p1}, \boldsymbol{V}_A^{c1}>$ and $s_{c2}=<\boldsymbol{V}_R^{p2}, \boldsymbol{V}_A^{c2}>$.

Due to the randomness of multi-point cross-

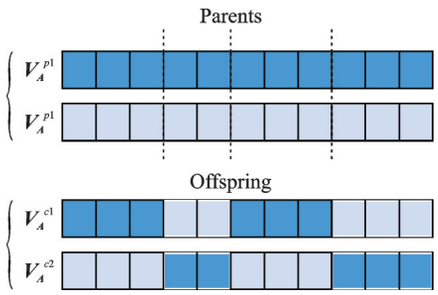

Fig.5 Multi-point crossover of the second chromosome

over, $< \boldsymbol{V}_R^{p1}, \boldsymbol{V}_A^{c1} >$ or $< \boldsymbol{V}_R^{p2}, \boldsymbol{V}_A^{c2} >$ could possibly become an illegal code. Since the first chromosome comes directly from a parent, it always satisfies constraint $C'1$; however, the second chromosome produced by the crossover may not satisfy $C'2$, which means there are tasks assigned to the rotation angles that do not exist in the first chromosome. Define the illegal angles in $\boldsymbol{V}_A^{c1}$ (that do not exist in $\boldsymbol{V}_R^{p1}$) is defined as

$$\boldsymbol{R}_{\text{illegal}}^{c1} = \left\{ r^{(i)} | r^{(i)} \in \boldsymbol{V}_A^{c1}, r^{(i)} \notin \boldsymbol{V}_R^{p1} \right\} \tag{11}$$

Due to the complexity of the problem and the randomness of crossover, there is a high probability of the occurrence of illegal codes. If the illegal codes are abandoned directly, the efficiency of the propagation process will be reduced. But more importantly, the offspring will mostly come from the crossover of similar parents, and this will obviously affect the performance of the algorithm. Therefore, we design the repair operator that fixes the illegal code while preserving the genetic information of its two parents as much as possible.

4.2.2 Repair operator

The repair operator is used to fix every illegal code derived from random crossover. Since the genetic information s_{c1} acquired from its parent s_{p2} all located in the second chromosome $\boldsymbol{V}_A^{c1}$, the repair operator should place as much illegal angles in $\boldsymbol{R}_{\text{illegal}}^{c1}$ into the first chromosome $\boldsymbol{V}_R^{p1}$ of s_{c1}, so that the information of s_{p2} can be preserved in s_{c1}. Any illegal angles may not be placed into the Null slots of $\boldsymbol{V}_R^{p1}$, because this will increase the number of rotations m, and cause that the repaired code always has a larger m value than the parents. Therefore, the illegal angles shall be placed into $\boldsymbol{V}_R^{p1}$ by replacing the existing angles in $\boldsymbol{V}_R^{p1}$. For $r^{(x)} \in \boldsymbol{R}_{\text{illegal}}^{c1}$, the placing by replacing process is as follows:

(1) Traverse through the existing angles in $\boldsymbol{V}_R^{p1}$ to find every candidate one that can be replaced by $r^{(x)}$ without violating $C'1$.

(2) Randomly chose a candidate angle and replace it with $r^{(x)}$ with a given probability ρ_{rpl}, if any candidate exist.

(3) If $r^{(x)}$ successfully replaces the existing angle r_j, $\boldsymbol{V}_R^{p1}$ turns into $\tilde{\boldsymbol{V}}_R^{p1}$, and the tasks assigned to r_j have to be reassigned.

(4) If $r^{(x)}$ fails to replace any existing angle, the tasks assigned to it have to be reassigned to the existing angles.

The repair process is shown in Fig.6, where $r^{(2)}$ is an illegal angle in offspring s_{c1}, and the repair operator uses it to replace the existing angle r_2; task t_j is once assigned to r_2 (before the replacement, $r^{(j)} = r_2$). After the replacement, it is reassigned to a new angle, and it has to be noted that the new angle does not necessarily be $r^{(2)}$.

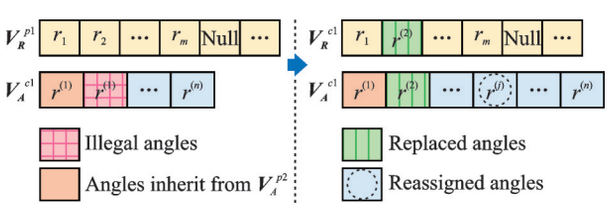

Fig.6 A demonstration of the repair operator

According to steps (1—4), there will be tasks in $\boldsymbol{V}_A^{c1}$ that need to be reassigned whether or not the replacement occurs: If the replacement occurs, the tasks once assigned to the original angle which has been replaced have to be reassigned; if not, the tasks that try to use the illegal angle have to be reassigned to existing angles. Here, the reassignment always selects the angle in $\tilde{\boldsymbol{V}}_R^{p1}$ that achieves the highest stiffness index for the tasks. After the repair, $\tilde{\boldsymbol{V}}_R^{p1}$ becomes a feasible first chromosome for s_{c1}, denoted as $\boldsymbol{V}_R^{c1}$. Moreover, if any angle in $\boldsymbol{V}_R^{c1}$ becomes redundant, it has to be deleted from $\boldsymbol{V}_R^{c1}$.

4.2.3 Mutation operator

After crossover and repair, the mutation operator ensure that every offspring has a probability of ρ_{mtt} to mutate. Once the mutation occurs, every element in the first chromosome mutates with a probability of ρ_{ms}. If $r_i \in \boldsymbol{V}_R$ mutates, the mutation is performed by randomly changing the angle of r_i within

a limited range

$$\tilde{r}_i = r_i + \theta_{\text{step}} * \text{rand}_i(-s_{\max}, s_{\max}) \quad (12)$$

where $s_{\max}$ is the maximum step limit of r_i mutation ($s_{\max}$ is an integer); function $\text{rand}_i(x, y)$ returns a random integer from range $[x, y]$. Eq.(12) means that a mutated rotation angle will move forward or backward randomly and a mutation result is feasible if it does not repeat with other angles, and $\boldsymbol{V}_R$ still satisfies $C'1$. If no feasible mutation is found within limited attempts, the algorithm will abandon the mutation of r_i. Reassignment of tasks will be needed if any rotation angle mutates.

The mutation of the second chromosome $\boldsymbol{V}_A$ is similar to $\boldsymbol{V}_{Rt}$: Every task has a probability of ρ_{ms} to reselect the rotation angle of processing, i.e. every element $r^{(i)}$ in $\boldsymbol{V}_A$ has a probability to change into another angle in $\boldsymbol{V}_R$.

4.3 Initializing population and the main loop

The individuals (i.e. codes) with the same value of m are defined as the same type, then there are $m_{\_\text{UB}} - m_{\min} + 1$ types of individuals exist in the problem ($\underline{m}_{\min}$ is the minimum number of rotations needed for a feasible solution). When the population is initialized for the evolution, each type of individuals shall be included and has a same ratio in the population.

When the population propagates, the entire process, including bi-tournament selection, crossover, repair, and mutation is roughly the same as in classic NSGA-Ⅱ, but with two modifications:

(1) Identical offspring shall not exist in the population. After each round of propagation, the repeated individuals will be deleted.

(2) When the new population is constructed, the individuals with a crowding distance of 0 is removed from every non-dominant layer that is included in the new population, rather than only the last layer included.

These measures are used to prevent the continuous multiplification of identical codes in the population, and maintain the population's ability of finding new solutions. Finally, the algorithm stops after a pre-set number of evolutions and outputs the Pareto-optimal solutions as the final solution. The overall process is illustrated in Fig.7.

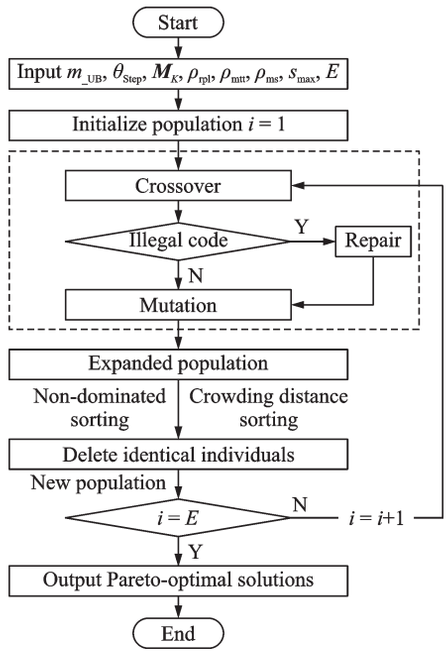

Fig.7 Flowchart of the algorithm

5 Experiments and Discussion

5.1 Machining experiment

A machining experiment is conducted on a test piece of the cabin to verify the correlation between the robot stiffness index k and the machining quality. Since k is concerned with the robot configuration $\boldsymbol{\theta}$, several machining processes are carried out using different configurations, and the machining quality is inspected and compared. Due to the limitation of experimental conditions, the test piece cannot be rotated like a real cabin. However, the robot can be repositioned using its mobile platform. Therefore, the change of $\boldsymbol{\theta}$ is achieved by shifting the robot between several base positions.

The layout of the experimental platform is shown in Fig.8. A KR210 industrial robot equipped with milling EE handles the machining of a support, which is mounted on the test piece. An accelerometer is attached to the bottom of EE to record the vibration. A total of seven base positions are used in the experiment, denoted as w_1—w_7, where w_4 directly faces the support, and the spacing between adjacent positions is 0.5 m, as shown in Fig.9.

The robot configurations and corresponding stiffness indices at positions w_1—w_7 are listed in Table 1. It can be found that k varies significantly with the base position. An 8 mm three-edge end mill is

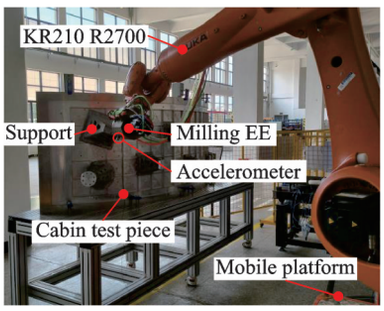

Fig.8 Machining experiment platform

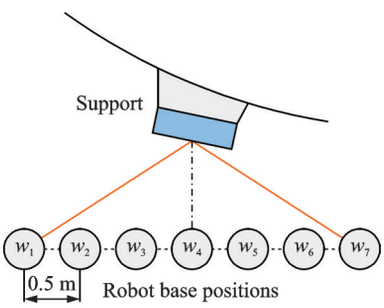

Fig.9 Robot base position setup of the experiment

used in the seven machining processes. The spindle speed is fixed at 5 000 r/min, and the robot feeds at a speed of 2.5 mm/s. The machined surfaces corresponding to the seven base positions are shown in Fig.10. The machining quality is verified through surface roughness, which is listed at the top of Fig.10. It can be found that the surface machined at w_3 has the minimum roughness (Ra0.353), while w_3 is also the position that has the highest k value (1 523 N/mm). The positions that have lower k values usually result in less desirable surfaces, e.g., k = 1 166 N/mm at w_1, and the roughness of the machined surface is Ra0.588.

Table 1 Robot configurations and corresponding stiffness indices at different positions

Position	θ_1/(°)	θ_2/(°)	θ_3/(°)	θ_4/(°)	θ_5/(°)	θ_6/(°)	k/(N•mm^{-1})
w_1	42.2	−60.7	75.6	160.2	−79.6	15.3	1 166
w_2	29.7	−75.2	95.8	158.9	−70.1	5.9	1 380
w_3	15.5	−85.2	107.4	158.7	−63.7	−7.0	1 523
w_4	−4.1	−89.4	108.3	161.6	−60.6	−27.6	1 506
w_5	−21.6	−84.6	103.8	165.7	−55.7	−46.4	1 403
w_6	−37.1	−72.6	89.6	171.3	−55.0	−64.3	1 199
w_7	−47.5	−57.2	67.5	175.6	−60.5	−77.0	967

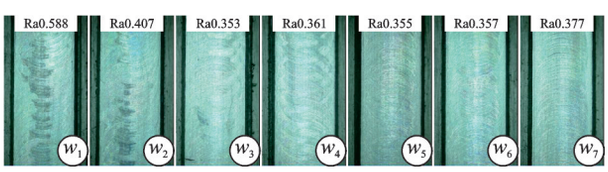

Fig.10 Machined surfaces of w_1—w_7

Apart from the inspection of the machined surface, the machining vibration is recorded by the accelerometer, and the maximum vibration acceleration in Cartesian space (denotes as a_{Crt}) at w_1—w_7 is shown in Fig.11. The k values in Table 1 are also visualized in Fig.11. A strong correlation can be found between a_{Crt} and k: From w_3 to other positions, a_{Crt} grows with the decrease of k. The results prove that optimizing the robot configuration to improve stiffness is helpful in reducing the machining vibration, hence more desirable machining quality can be expected.

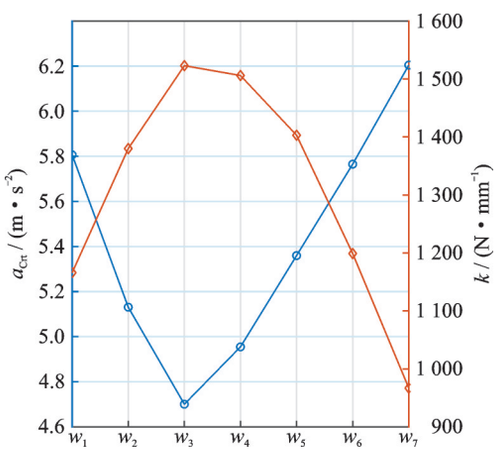

Fig.11 Correlation between the machining vibration (max Cartesian acceleration) and the robot stiffness index

Although the experiment is restricted in its form, it provides a verification of the positive correlation between the stiffness index k and the machining quality: Optimized configuration with higher stiffness will suppress machining vibration and improve the quality of machined surface. Therefore, the robot stiffness index k proposed in Section 2, as well as the WMSI index generated from it, provides reliable evaluations to the machining quality of a single machining task and the whole cabin. Moreover, the experiment proves that robot configuration $\boldsymbol{\theta}$ is the key to improve k, while $\boldsymbol{\theta}$ can either be optimized through the change of robot base position or through the rotation of the cabin.

5.2 Computational experiments

This section takes a simulated cabin with three cylindrical segments as a case study. The rotation schemes as well as the machining solutions to it are

optimized by the proposed method. The simulated cabin is 6.6 m long with a 3.6 m diameter. After been attached to the tooling system, the axis of the cabin is 2.2 m above the shop floor, and a total of 90 small supports are distributed on the surface of the cabin. These are the 90 machining tasks that need to be handled by the robots. Three KR500-3 robots are deployed on each side of the cabin, and each segment has two robots operating on its left and right sides. The position of the robot relative to the cabin is shown in Fig.12. The robots are restricted to process tasks located within the height ranging from 1.3 m to 3.1 m (covering about 60° of the cabin's circumference) so that they will not operate near the edge of their workspace. The base-to-base distance of adjacent robots is 2.2 m.

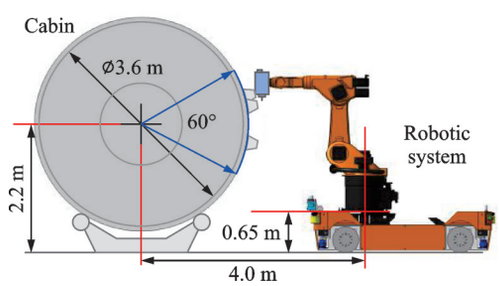

Fig.12 Position of the robot on one side of the cabin

Given $\theta_{step} = 10°$, the machining stiffness indices of the 90 tasks on all 36 rotation angles are calculated in advance, and the joint stiffness matrix $\boldsymbol{K}_\theta$ of KR500-3 adopts the results of a previous study[28]. The results form a 90×36 matrix, $\boldsymbol{M}_K$, where line i ($i = 1, 2, \cdots, 90$) is the stiffness indices of task t_i. If t_i cannot be processed by any robot at rotation angle r_j, the stiffness index $k_{i,j} = 0$. Task t_1 is taken as an example. It can be machined at 13 rotation angles (−100°——50°, and 80°—140°, corresponding to the two robots on each side of the cabin). The stiffness index k_1 is shown in Fig.13. It can be seen that k_1 changes significantly with the selection of rotation angle, and the maximum value is almost twice the minimum, and only on five angles k_1 will be above 1 500 N/mm, indicating that improving the machining quality of t_1 requires a proper selection of rotation angles.

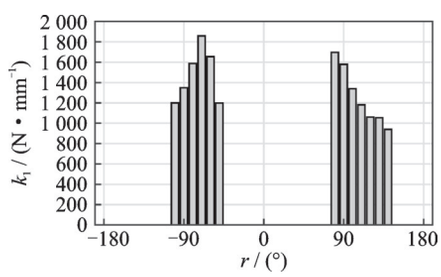

Fig.13 Machining stiffness index k_1 of task t_1 on different rotation angles

The proposed algorithm is used to optimize the machining solutions of the object cabin, and the parameter setup of the algorithm is listed in Table 2. A total of eight Pareto-optimal solutions are found by the algorithm, as shown in Fig.14. The minimum value of m, $m_{min} = 3$. That is, at least three rotations will be needed to complete all tasks. In Fig.14, when m grows from 3 to 10, WMSI also grows significantly, from 1 372.0 N/mm to 1 693.8 N/mm.

Table 2 Parameter settings of the proposed algorithm

Variable	Value	Explanation
θ_{step}/(°)	10	Step length of the rotation
$m_{\_UB}$	10	Maximum number of rotations
$[l_2, l_1, h_1, h_2]$	[1.3, 1.15, 0.85, 0.7]	Weighting factors of WMSI
ρ_{rpl}	0.70	Parameter of repair operator
$[\rho_{mtt}, \rho_{ms}, s_{max}]$	[0.10, 0.15, 6]	Parameters of mutation operator
$[N, E]$	[100, 200]	Population scale and number of evolutions of NSGA-Ⅱ

To intuitively analyze to what extent the eight solutions are close to the quality optimal solution, we define the optimization level of machining quality

$$q_{opt} = \mathrm{WMSI}/\mathrm{WMSI}_{max} \tag{13}$$

where WMSI_{max} is the maximum value that WMSI can achieve in the problem. That is, when each task is assigned to the rotation angle that maximizes its machining stiffness (e.g., assign t_1 to −70°). In the

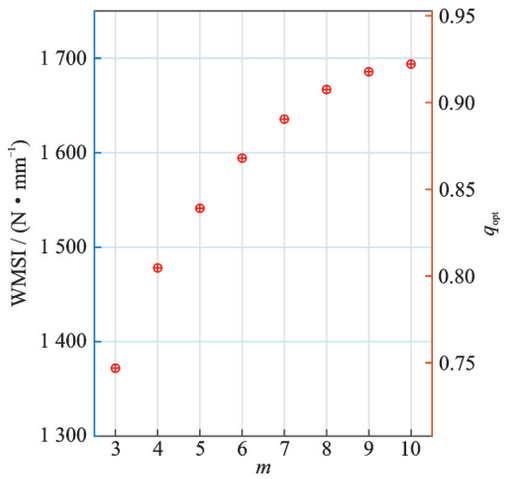

Fig.14 Pareto-optimal solutions

studied case, $WMSI_{max} = 1\ 836.9$ N/mm, but up to 36 rotations will be required to achieve this, which is almost impractical in real-world applications. In Fig.14, when $m = 3$, $q_{opt} = 0.747$, while $m = 10$, $q_{opt} = 0.922$. Such results are already quite close to the quality-optimal solution. Moreover, we notice that neither WMSI nor q_{opt} grows linearly with m. When m grows from 3 to 6, q_{opt} increases by 0.121; when m further increases to 10, q_{opt} only increases by 0.054. This indicates that the effect of increasing the number of rotations on improving the machining quality is gradually reduced, and in engineering practice, properly selecting the value of m will help balance the goal of machining quality and productivity. The indices of the eight solutions are listed in Table 3.

Table 3 Indices of the eight solutions

m	WMSI/(N•mm^{-1})	q_{opt}	Increment of q_{opt}
3	1 372.0	0.747	—
4	1 478.1	0.805	0.058
5	1 541.2	0.839	0.034
6	1 594.3	0.868	0.029
7	1 635.5	0.890	0.022
8	1 666.9	0.907	0.017
9	1 685.7	0.918	0.011
10	1 693.8	0.922	0.004

In Fig.15, eight boxplots are used to visualize the distribution of 90 stiffness indices $(k_1, k_2, \cdots, k_{90})$ in the eight solutions. In a boxplot, the horizontal line inside the box denotes the median value of the data set, while the top and the bottom of the box denote the upper and the lower quartiles, respectively. It can be found that with the increase of m, the stiffness indices of the 90 tasks have an overall increase, and almost all the tasks have their stiffness index above 1 400 N/mm when $m \geqslant 8$.

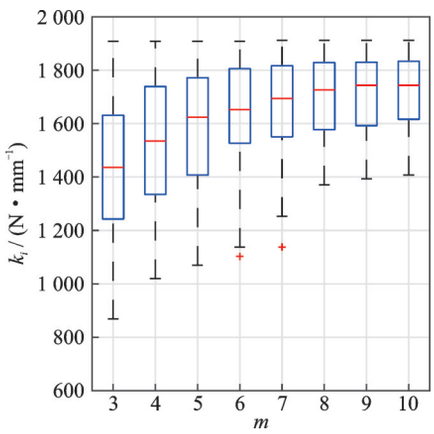

Fig.15 Stiffness index distribution of the 90 tasks (f_2=WMSI)

To verify whether the proposed index WMSI helps bias the algorithm towards improving the lower values in the 90 indices, the mean stiffness index (MSI) is introduced by changing the weights in WMSI into $l_2 = l_1 = h_1 = h_2 = 1$. To statistically compare the results generated using WMSI and MSI, 50 random instances of task distribution are generated. That is, the positions of the 90 small supports on the cabin are randomly changed 50 times. The algorithm is executed 50 times with different instances using WMSI and MSI seperately. The comparison is conducted between the lower k values (i.e. $\boldsymbol{K}_{l_1} \cup \boldsymbol{K}_{l_2}$). Fig.16 analyses the distribution of the lower k values with the mean value, the standard deviation (S.D.), and the minimum value, where the solid line indicates the median, the upper and the lower edges of the shadow block indicate the upper and lower quartiles, respectively. Except for the results with $m = m_{min} = 3$, compared to MSI, WMSI let the lower k values have higher mean and minimum values, as well as lower S.D., indicating that WMSI can effectively bias the algorithm toward improving the lower k values, thus more reasonable solutions will be produced by using the WMSI index.

To verify the efficiency of the proposed algorithm, we compare the optimized solutions produced by the algorithm with the solutions currently used in engineering practice (denoted as EP solu-

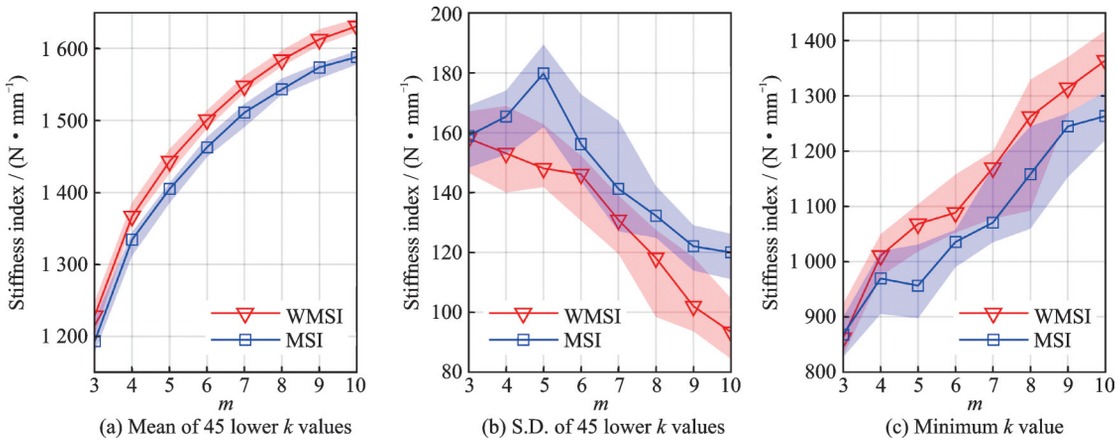

Fig.16 Comparison of the lower k value distribution between the results produced using WMSI and MSI

tions). In engineering practice, the cabin is rotated with a fixed angle, e.g., if $m = 6$, the cabin rotates 60° each time. Each task will then be assigned to the angle that maximizes the robot stiffness index. When $m < 6$, such method does not find rotation schemes that satisfy constraint $C'1$, therefore only five EP solutions ($m = 6$—10) can be compared with the corresponding optimized solutions. The comparison with respect to WMSI is shown in Fig.17.

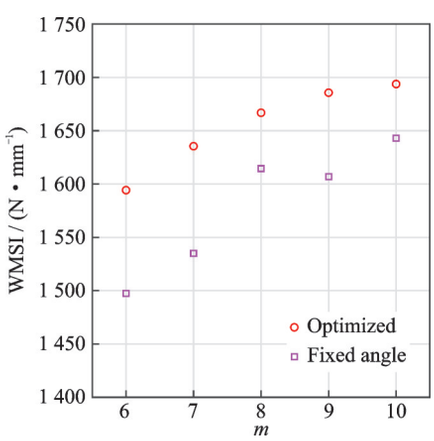

Fig.17 Comparison between the optimized solutions and EP solutions

It is obvious that the optimized solutions outperform the EP solutions, and it is interesting to find that when m increases from 8 to 9, the value of WMSI decreases for the EP solutions. This indicates that a proper rotation scheme (rather than just more rotation angles) is quite essential to improving the machining quality of the whole cabin. In fact, when $m = 9$, the rotation scheme proposed by the algorithm is [30°, 60°, 80°, 110°, 130°, 160°, 210°, 260°, 350°], while the rotation scheme proposed by the EP method (i.e., fixed angle) is [0°, 40°, 80°, 120°, 160°, 200°, 240°, 280°, 320°] (maximum WMSI can be achieved when the rotation starts from 0°). The two schemes are obviously different.

The optimized solutions can also gain advantages in production efficiency by reducing the rotations required for the cabin: When $m = 6$, the optimized solution achieves the WMSI index of 1 594.3 N/mm, while for the EP solution, two more rotations are required to achieve the WMSI index no worse than the former (1 615 N/mm). When $m = 7$, the WMSI index of optimized solution is 1 635.5 N/mm, only the EP solution with $m = 10$ can compete it (WMSI = 1 643 N/mm). Since less rotations can reduce interruptions and recalibration workload of the whole process, more desirable efficiency can be obtained by the proposed method without affecting the machining quality.

6 Conclusions

This paper studies a dual-objective optimization problem of production efficiency and machining quality in the in-situ manufacturing of a large spacecraft cabin. The production efficiency is dominated by the number of rotations performed to the cabin, and the machining quality is mainly affected by the robot stiffness in machining, which is also influenced by the rotation of the cabin. A solution that specifies the optimal rotation scheme of the target cabin is required for the manufacturing mission.

The robot stiffness index is introduced as a ma-

chining quality evaluation of a single task. The correlation between the index and the machining quality is then experimentally verified. To properly evaluate the machining quality of the whole cabin, a stiffness-based global evaluation index, WMSI, is further proposed. The main contribution of the study is the weighting strategy applied to WMSI and the dual-objective optimization algorithm designed for the problem. Results of the computational experiment show that the algorithm provides more desirable solutions than currently used EP solutions, both in machining quality and in production efficiency. Further, the proposed WMSI index biases the algorithm towards improving the lower stiffness indices among the tasks, hence more reasonable solutions will be produced.

Future research may focus on balancing the workload among the robots in order to equalize tool wear and uniform tool replacement time, so that online break down can be avoided as much as possible.

References

[1] ALLEN J, AXINTE D, ROBERTS P, et al. A review of recent developments in the design of special-purpose machine tools with a view to identification of solutions for portable in situ machining systems[J]. The International Journal of Advanced Manufacturing Technology, 2010, 50(9): 843-857.

[2] SUSEMIHL H, MOELLER C, KOTHE S, et al. High accuracy mobile robotic system for machining of large aircraft components[J]. SAE International Journal of Aerospace, 2016, 9(2): 231-238.

[3] MOELLER C, SCHMIDT H C, KOCH P, et al. Machining of large scaled CFRP-parts with mobile CNC-based robotic system in aerospace industry[J]. Procedia Manufacturing, 2017, 14: 17-29.

[4] ZHANG L, TIAN W, ZHENG F, et al. Accuracy compensation technology of closed-loop feedback of industrial robot joints[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(6): 858-871.

[5] LJASENKO S, FERREIRA P, JUSTHAM L, et al. Decentralised vs partially centralised self-organisation model for mobile robots in large structure assembly[J]. Computers in Industry, 2019, 104: 141-154.

[6] LJASENKO S, LOHSE N, JUSTHAM L. A comparison of the manufacturing resilience between fixed automation systems and mobile robots in large structure assembly[J]. Procedia Cirp, 2016, 57: 235-240.

[7] PAN Z, ZHANG H, ZHU Z, et al. Chatter analysis of robotic machining process[J]. Journal of Materials Processing Technology, 2006, 173(3): 301-309.

[8] TAO B, ZHAO X W, DING H. Mobile-robotic machining for large complex components: A review study[J]. Science China Technological Sciences, 2019, 62(8): 1388-1400.

[9] VERL A, VALENTE A, MELKOTE S, et al. Robots in machining[J]. CIRP Annals, 2019, 68(2): 799-822.

[10] ABELE E, WEIGOLD M, ROTHENBUCHER S. Modeling and identification of an industrial robot for machining applications[J]. CIRP Annals, 2007, 56(1): 387-390.

[11] GONUL B, SAPMAZ O F, TUNC L T. Improved stable conditions in robotic milling by kinematic redundancy[J]. Procedia CIRP, 2019, 82: 485-490.

[12] DILLHOEFER T. Power RACe[J]. SAE Technical Paper, 2017, 1: 2093.

[13] BU Y, LIAO W, TIAN W, et al. Stiffness analysis and optimization in robotic drilling application[J]. Precision Engineering, 2017, 49: 388-400.

[14] BU Y, LIAO W, TIAN W, et al. Modeling and experimental investigation of Cartesian compliance characterization for drilling robot[J]. The International Journal of Advanced Manufacturing Technology, 2017, 91(9): 3253-3264.

[15] GUO Y, DONG H, KE Y. Stiffness-oriented posture optimization in robotic machining applications[J]. Robotics and Computer-Integrated Manufacturing, 2015, 35: 69-76.

[16] JIAO J, TIAN W, LIAO W, et al. Processing configuration off-line optimization for functionally redundant robotic drilling tasks[J]. Robotics and Autonomous Systems, 2018, 110: 112-123.

[17] XIONG G, DING Y, ZHU L M. Stiffness-based pose optimization of an industrial robot for five-axis milling[J]. Robotics and Computer-Integrated Manufacturing, 2019, 55: 19-28.

[18] CHEN C, PENG F, YAN R, et al. Stiffness performance index based posture and feed orientation optimization in robotic milling process[J]. Robotics and Computer-Integrated Manufacturing, 2019, 55: 29-40.

[19] FAN Q, GONG Z, TAO B, et al. Base position optimization of mobile manipulators for machining large complex components[J]. Robotics and Computer-Integrated Manufacturing, 2021, 70: 102138.

[20] MUTTI S, NICOLA G, BESCHI M, et al. Towards optimal task positioning in multi-robot cells, using nested meta-heuristic swarm algorithms[J]. Robotics and Computer-Integrated Manufacturing, 2021, 71: 102131.

[21] SPENSIERI D, CARLSON J S, BOHLIN R, et al. Optimal robot placement for tasks execution[J]. Procedia CIRP, 2016, 44: 395-400.

[22] LIM Z Y, PONNAMBALAM S G, IZUI K. Multi-objective hybrid algorithms for layout optimization in multi-robot cellular manufacturing systems[J]. Knowledge-Based Systems, 2017, 120: 87-98.
[23] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
[24] SHAO Z, HE C, PEI J. Multi-objective optimization design of vented cylindrical airbag cushioning system for unmanned aerial vehicle[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2016, 33(2): 208-214.
[25] SALISBURY J K. Active stiffness control of a manipulator in Cartesian coordinates[C]//Proceedings of the 19th IEEE Conference on Decision and Control Including the Symposium on Adaptive Processes. Albuquerque, USA: IEEE, 1980: 95-100.
[26] CORDES M, HINTZE W. Offline simulation of path deviation due to joint compliance and hysteresis for robot machining[J]. The International Journal of Advanced Manufacturing Technology, 2017, 90(1): 1075-1083.
[27] WANG G, DONG H, GUO Y, et al. Dynamic cutting force modeling and experimental study of industrial robotic boring[J]. The International Journal of Advanced Manufacturing Technology, 2016, 86(1): 179-190.
[28] JIAO J, TIAN W, ZHANG L, et al. Variable parameters stiffness identification and modeling for positional compensation of industrial robots[J]. Journal of Physics, 2020, 1487(1): 012046.

Acknowledgement This work was supported in part by the National Natural Science Foundation of China (No. 52075256). The authors realize that the time and space available for a review of such an ambitious subject are limited and, thus, regretfully, we are unable to cover many important contributions.

Authors Mr. **LIU Shaorui** received the B.S. degree in aircraft manufacturing engineering from the North University of China, Taiyuan, China, and the M.S. degree in aeronautical engineering from the College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2014 and 2018, respectively. He is currently a Ph.D. candidate in aeronautical and astronautical manufacturing engineering at NUAA. His research interests are multi-robot systems, planning algorithms and aircraft assembly technologies.

Prof. **TIAN Wei** received his B.S., M.S., and Ph.D. degrees from the Nanjing University of Science and Technology in 2000, 2003, and 2006, respectively. He is now a professor of aeronautical and astronautical manufacturing engineering with Nanjing University of Aeronautics and Astronautics, and the leader of the Lab of Aerospace Robot Intelligent Assembly. His research interests are digital and flexible assembly technologies in aerospace manufacturing and high performance robotic systems.

Author contributions Mr. **LIU Shaorui** conducted modeling, algorithm design, experimentation, data processing and analysis, and wrote the original manuscript. Prof. **TIAN Wei** contributed to conceptualization, modeling, resources, and project administration, and wrote the review and edited the manuscript. Prof. **SHEN Jianxin** contributed to the conceptualization, investigation, formal analysis, and project administration, and wrote the review and edited the manuscript. Dr. **LI Bo** guided the experimentation, and contributed to formal analysis and data processing. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

面向机器人原位加工的大型飞船舱体变位姿态优化方法

刘少睿，田　威，沈建新，李　波
（南京航空航天大学机电学院，南京 210016，中国）

摘要：在利用机器人实施大型飞船舱体的原位加工时，对舱体旋转变位次数与机器人整体加工性能的关注引入了一双目标优化问题。本文基于机器人刚度特性和非支配排序遗传算法，提出了一种舱体变位方案优化方法。首先，设计了以机器人笛卡尔刚度为基础的特征加工质量评价指标，并建立了原位加工过程模型。其次，为应用非支配排序遗传算法，提出了一种加工过程的双染色体编码方法及相应的交叉变异算子。此外，通过修复算子处理频繁出现的非法编码，保证了算法的寻优效率。仿真及实验研究的结果表明，适当增加舱体变位次数能够有效提升机器人的加工性能，并在工质量和时间成本上实现综合优化。

关键字：大型结构件制造；原位加工；机器人刚度优化；双目标优化；非支配排序遗传算法