

A tool path optimization approach based on blend feature simplification for multi-cavity machining of complex parts

Science Progress

1–21

© The Author(s) 2019

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/0036850419874233

journals.sagepub.com/home/sci

Yupeng Xin^{1,2}, Shengqiang Yang¹, Gangfeng Wang² ,
Richard Evans³, and Fengfeng Wu¹

¹College of Mechanical and Vehicle Engineering, Taiyuan University of Technology, Taiyuan, China

²Key Laboratory of Road Construction Technology and Equipment of MOE, School of Construction Machinery, Chang'an University, Xi'an, China

³College of Engineering, Design and Physical Sciences, Brunel University London, London, UK

Abstract

Blend features usually exist in the machining of complex multi-cavity parts; however, the ideal linear boundary of the cavity is shown as an arc curve at actual corner machining, which affects the accuracy of a robot's tool feed position. Focused on this problem, this article presents an automatic tool path planning approach based on blend feature simplification. By analyzing the geometric elements of blend feature, a line segment is constructed to obtain the machining boundary, while the robot tool feed position is accurately measured. On this basis, the coordinates of a robot tool feed position are assigned to the machining element, which can be used to calculate the spatial distance between different cavities. Then, an improved genetic algorithm is applied to improve the optimization of the tool path. The automatic decision of the corresponding work steps is realized by merging and sorting the machining elements. Finally, a corresponding prototype system is presented, with the correctness and validity of the proposed approach being examined, using aircraft structural part machining as an illustrative example.

Keywords

Smart process planning, tool path optimization, blend feature, multi-cavity machining, digital manufacturing

Corresponding authors:

Yupeng Xin, College of Mechanical and Vehicle Engineering, Taiyuan University of Technology, Taiyuan 030024, China.

Email: xinyupeng@tyut.edu.cn

Gangfeng Wang, Key Laboratory of Road Construction Technology and Equipment of MOE, School of Construction Machinery, Chang'an University, Xi'an 710064, China.

Email: wanggf@chd.edu.cn



Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (<http://www.creativecommons.org/licenses/by-nc/4.0/>)

which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

Introduction

A cavity is a common structure in an aircraft structural and mold design. It is usually processed by a robot or computer numerical control (CNC) milling.¹⁻³ Aircraft structural parts are usually large in size and multi-cavity (the overall size can reach 4000 mm × 2000 mm); in this instance, cavities tend to be far apart from each other. In addition, the robot tool path planning of multi-cavity parts must consider each cavity's profile and its spatial position relationship, keeping the machining mode unchanged and minimizing the number of tool changes required.^{4,5}

The robot tool path optimization is often summarized as the traveling salesman problem (TSP).^{6,7} These recognized problems often use heuristic algorithms to obtain approximate solutions, the most common of which are genetic algorithms (GAs), ant colony algorithms, and particle swarm algorithms.⁸⁻¹² Li et al.¹³ used a multi-objective optimization approach, based on neural networks to solve the problem of tool path planning during the engraving process. First, a back propagation neural network model was constructed. Then, energy consumption and the machined surface roughness were identified as the constraint conditions, with the trajectory spacing and feature machining time being used as input parameters. The optimization results of the machining parameters and the machining trajectory were obtained by training. Although this method considers the spatial position of the machining feature, the distance between the features depends on manual calculation. The calculation result is input as a parameter of the optimization algorithm, which not only has a low degree of intelligence but also makes it difficult to guarantee the accuracy of the calculation result.

Chu et al.¹⁴ proposed a five-axis milling tool path planning approach based on an ant colony optimization algorithm for a five-axis CNC milling machine. This trajectory planning problem was specified as two boundary curves for drawing a regular surface. The authors used the algorithm to search for a mapping that minimizes machining errors. The division of the boundary greatly reduced the sample space, resulting in a significant increase in computational efficiency. However, in the actual machining, the feature boundary of the part was affected by the blend feature (arc surface or trapezoidal angular), which is often difficult to obtain directly. Therefore, the application of this approach is deemed to have significant limitations. In addition to the aforementioned, Kuo et al.¹⁵ attempted to use the algorithm based on an electromagnetic mechanism (EM) for tool path planning. Their study showed that an EM-based algorithm can produce an optimal solution more efficiently than a heuristic algorithm, even though the search time is longer. Plakhotnik and Lauwers¹⁶ designed a tool path optimization algorithm to find the shortest trajectory by solving the entire tool path minimization cost function, including machine rotation axis displacement, but the method could not handle the non-linear function.

Although the aforementioned research has used different methods to achieve the optimization of tool paths, for multi-cavity parts with blend features, the optimization results cannot be directly applied. Secondary processing of the blend features is therefore required. In the actual machining of multi-cavity parts, to meet safety and machinability requirements and avoid cracks in the corners of the cavity due to stress, fillets are commonly used instead of sharp edges or sharp corners to form blended features. The

numerical control machining of the out-of-profile contours can be achieved by simply modifying the tool's offset parameters. The blend features of the out-of-cavity profile can be achieved by simply modifying the tool's offset parameters in the robot machining. However, when machining internal contours, the programmer must manually add an auxiliary transition arc and ensure that the transition arc radius is greater than the tool radius. This not only increases the difficulty in numerical control programming but also neglects the part of the programmer which may result in the transition radius being smaller than the tool radius, meaning that the machined part will be scrapped due to the interference caused by tool cutting.

In order to simplify the calculation process and improve the programming efficiency of robot machining of multi-cavity parts, scholars have proposed suppressing the blend feature. The idea being to replace the arc face of the blend feature with a new face and to ensure that the new face is always tangent to the supportive surface by increasing constraints.^{17–20} These methods require the reconstructing of the three-dimensional (3D) design model, with the fillet feature being suppressed to edge or point. To ensure the topological and geometric validity of the modified 3D models, it is necessary to change the topology and geometric information of the design models. However, it is forbidden to change the 3D design models at the robot programming stage. This idea does not apply to the robot programming process of multi-cavity parts.

Hence, our main objective in this research is to overcome the aforementioned problem. We propose a robot tool path planning approach based on blend feature simplification (BFS). A novel machining element is defined as the minimum information entity containing the decision information of work steps. By matching the cavity and machining elements, we can calculate the spatial distance between different machining features. Finally, the optimal robot tool path is obtained by optimizing the work steps based on an improved GA. The architecture used in this article is shown in Figure 1.

This article focuses on the influence of blend features on the robot tool path planning of multi-cavity parts. The rest of this article is structured as follows: section “Geometrical background” describes the geometrical background. Algorithm modeling of BFS is presented in section “Algorithm modeling of BFS.” In section “Robot tool path optimization using BFS,” we describe an improved-GA algorithm, which is used to optimize the tool path of multiple cavities. Section “Case study” contains a case study and discussions to prove the proposed method. The article ends with conclusions and identified areas for future research.

Geometrical background

Cavity features are commonly found in aircraft structures and mold designs, which are usually formed by four cavity walls and a cavity bottom. An aircraft's fuselage frame is a typical multi-cavity structure. To satisfy safety and machinability requirements, and to avoid corner stress, which is generated by the concentration of cracks, blend features are used to replace sharp edges or sharp corners, as shown in Figure 2.

In general, blend features are the intersection of two planes or multiple planes, using arc surfaces instead of their intersections. Blend features usually include a cylindrical surface, spherical surface, ring surface, and B-spline surface, as shown in Figure 3.

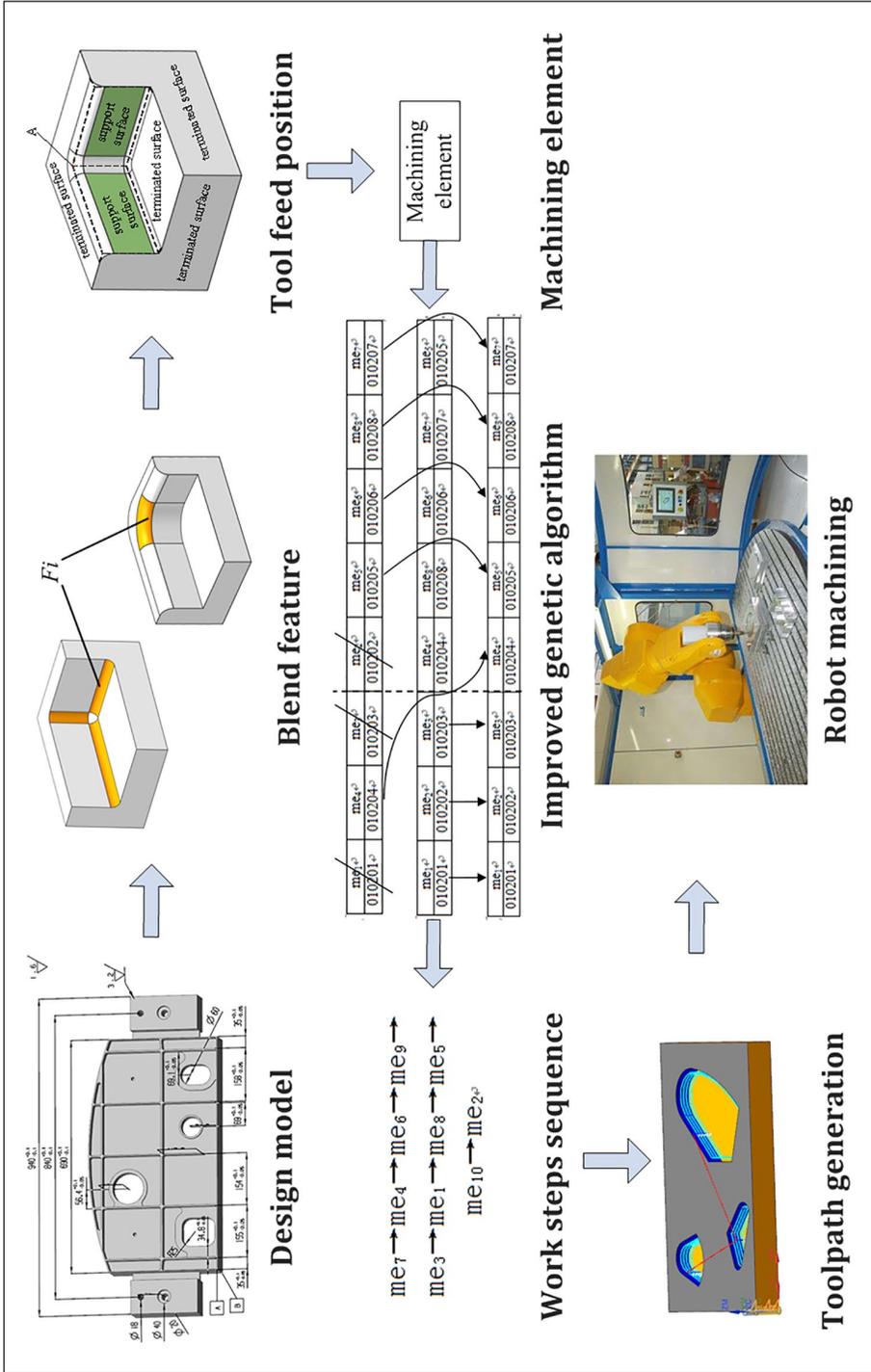


Figure 1. Flow chart of robot tool path planning based on BFS.

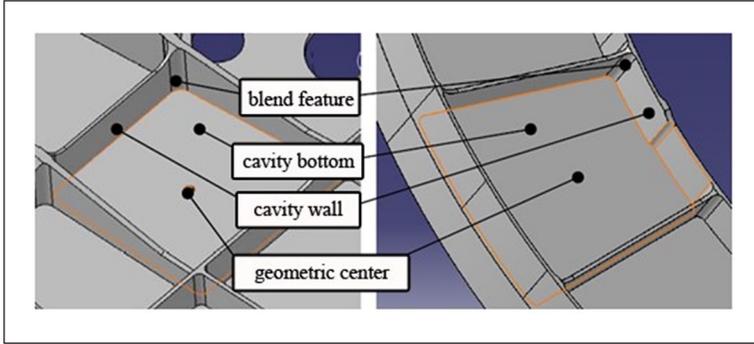


Figure 2. Blend features in aircraft cavity part.

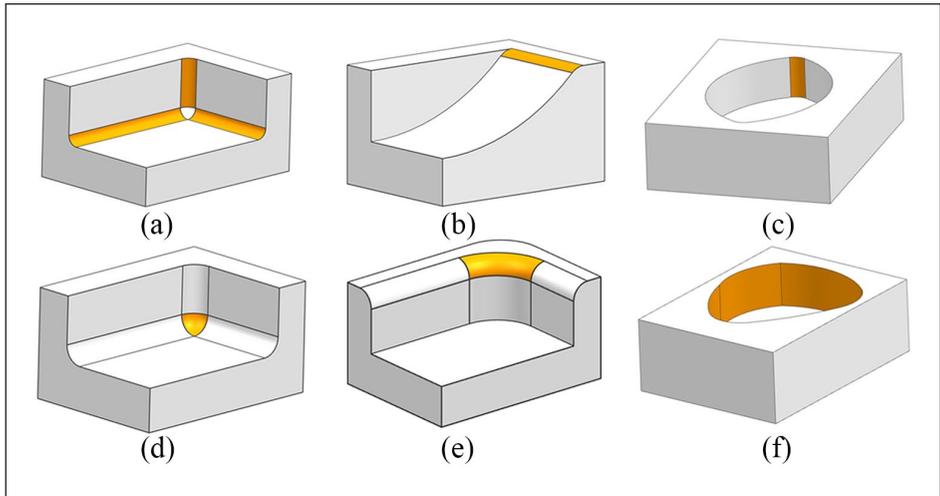


Figure 3. Blend feature types: (a) cylindrical surface, (b) cylindrical surface, (c) cylindrical surface, (d) spherical surface, (e) ring surface, and (f) B-spline surface.

The existence of the blend feature changes the original cavity structure, and the original linear boundary becomes a curved surface, which makes it difficult to obtain the coordinates of the tool feed position. The key to eliminating blend feature interference is to reconstruct the linear boundary of the cavity without changing the geometric elements of the blend feature. Different types of blend features contain different geometric elements (points, lines, surfaces, etc.). Table 1 describes the geometric elements of different types of blend features.

Transitional surface refers to the smooth arc surface of the blend feature, including transitional surface of the boundary type and transitional surface of the point type. Boundary transitional surface refers to the curved surface formed by rounding the sharp edge. Point transitional surface is an arc surface formed by the intersection of two or

Table 1. Geometric elements of blend feature.

Type	Description of geometric elements
Cylindrical surface	Blend connection between two planes (Figure 3(a)) Blend connection between a plane and a curved surface (Figure 3(b)) Blend connection between two cylindrical surfaces which have the same axial direction (Figure 3(c))
Spherical surface	Blend connection between three adjacent cylindrical surfaces, with a boundary of three arcs and a radial angle of three arcs less than 180° (Figure 3(d))
Circle surface	Blend connection between two cylindrical surfaces, usually the boundary is four arc edges and three cylindrical faces in adjacent faces (Figure 3(e))
B-spline curve surface	A set of surfaces that approximate the smooth curve of the control curve (Figure 3(f))

more edged transitional surfaces, which are used to replace the point of intersection formed by the intersection of the sharp edges, as shown in Figure 4(a).

The key to successful cavity tool path planning is to obtain the machining feature boundary, which includes the geometric elements shown in Figure 4(b). The supportive surface of the blend feature contains the boundary line of the blend feature, which is contiguous with the arc face. The terminate surface of the blend feature contains the sharp edge of the arc of the blend feature and it is contiguous with the arc face. The boundary line of the blend feature is the curve formed by the extension of the supportive surface, denoted as \mathcal{L} . If \mathcal{L} is closed, then \mathcal{L} is a simplified transitional curve representing the true machining boundary. Otherwise, \mathcal{L} is a curve representing the direction of the machining boundary.

Algorithm modeling of BFS

From a geometric point of view, when we process the cavity in the machining allowance that has been determined, the calculation of the tool feed position is the key to robot programming. Due to the interference of blend features, in the actual calculation process, we must calculate the tool feed position (point A, as shown in Figure 4(b)) according to the blend feature's radius. In order to simplify the calculation process, and accurately obtain the coordinates of point A, we reconstruct the blend feature's line segment \mathcal{L} , which can be divided into two categories: (1) line segment of the boundary blend feature and (2) line segment of the point blend feature. BFS can also be divided into two types.

BFS for boundary blend feature

The supportive surfaces of the blend feature are extended to intersect. After the intersection is complete, the terminated surface is truncated to obtain \mathcal{L} of the blend feature. Due to supportive surface loss (Figure 3(b) and (c)) or fading (Figure 3(f)) often occurring in

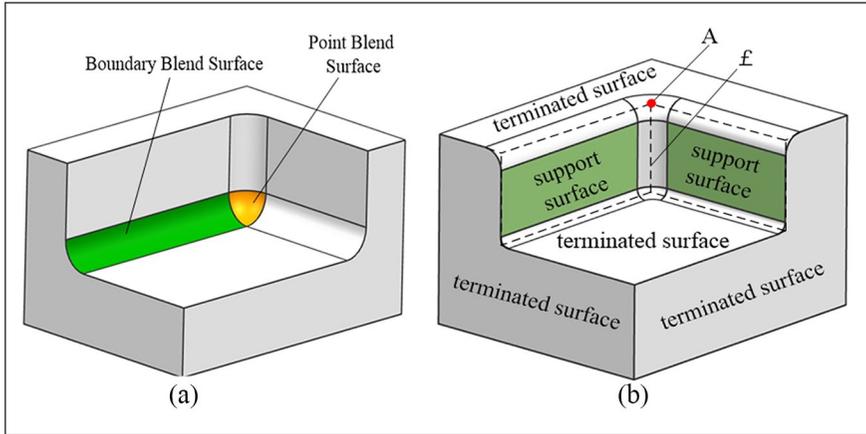


Figure 4. Geometric elements of blend feature: (a) blend surface and (b) geometric elements.

blend features, it is more difficult to reconstruct \mathcal{L} . Therefore, the supportive surface is replaced by an auxiliary plane.

BFS for point blend feature

From the viewpoint of boundary changes, no matter how complicated the topological boundary of the point blend feature is, the extension lines of its related edges on the surrounding supportive surfaces are always intersected. Based on this principle, the feature line segment of the point blend feature's surface is mainly obtained by extending the corresponding feature line segments of the adjacent side transition surface to intersect. A flow chart depicting this is shown in Figure 5. The reconstruction algorithm for \mathcal{L} is as follows:

- Step 1.* Select the blend feature in the 3D geometric model.
- Step 2.* If the blend feature has point transitional surfaces, then execute Step 3. If the blend feature has boundary transitional surfaces, execute Step 4.
- Step 3.* Get all transitional surfaces of point blend features.
- Step 4.* If the supportive surface of the boundary blend feature does not exist, execute Step 5. If the supportive surface of boundary blend feature exists, execute Step 6.
- Step 5.* Build auxiliary planes.
- Step 6.* Extend the supportive surfaces to intersection, and the resulting intersection is the blend feature line \mathcal{L} , as shown in Figure 4(b).
- Step 7.* If the blend feature line is a closed curve, then the blend feature line segment is \mathcal{L} . If the blend feature line is a non-closed curve, execute Step 8.

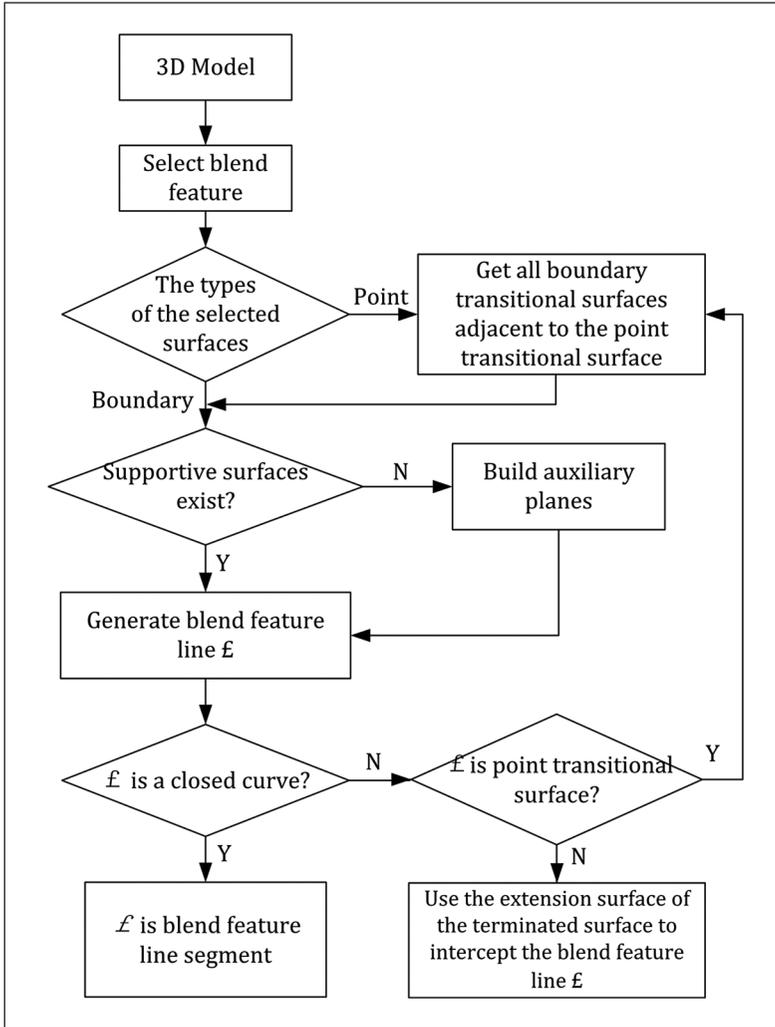


Figure 5. Flow chart of the blend feature line segment reconstruction.

Step 8. If the adjacent surface of the blend feature line is the point transitional surface, go to Step 9. If the adjacent surface of the blend feature line is not point transitional surface, return to Step 3.

Step 9. Use the extension surface of the terminated surface to intercept the blend feature line ℓ to form the blend feature line segment ℓ , as shown in Figure 4(b).

Step 10. The intersection of the terminated surface and ℓ is point A, as shown in Figure 4(b).

The steps involved in Step 5 to build the auxiliary plane are as follows:

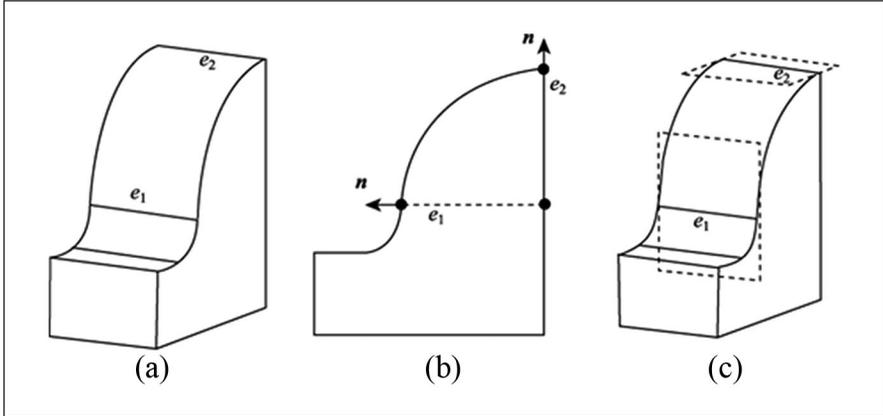


Figure 6. Auxiliary plane construction: (a) select datum boundary, (b) vector n of the auxiliary surface, and (c) construct auxiliary plane.

Step 5.1. Select e_1 and e_2 of the boundary transitional surface, as shown in Figure 6(a), as the reference boundary.

Step 5.2. The surface normal of the transitional surface at any point on the boundary e is the surface normal of the supportive surface of e , and the normal vector must cross the corresponding centroid of the transitional surface to determine the surface normal vector n of the auxiliary surface, as shown in Figure 6(b).

Step 5.3. Reconstruct the missing support plane using the reference boundary e and the surface normal n , as shown in Figure 6(c).

Based on the BFS, we can obtain the coordinates of the robot tool feed point (Point A, as shown in Figure 4(b)), which can be used to inform the automatic decision-making and optimization of the multi-cavity work step sequence.

Robot tool path optimization using BFS

In the machining process of multi-cavity parts, we can obtain the coordinates of tool feed position using BFS. On this basis, if we calculate the optimal path between different tool feed positions, the optimal robot tool path can be determined. To realize the automatic decision-making and tool path optimization for multi-cavities, we define the concept of a machining element. A machining element is the smallest unit that describes the machining of features. This includes information related to features, methods, manufacturing resources (such as robots, fixtures, and measuring tools), cutting parameters, and tool movement mode. The mathematical expression for the machining element can be expressed as

$$me_i = \{f_i, MP_i, MR_i, TAD, A_i(x, y, z)\} \quad (1)$$

where f_i is the machining feature, MP_i is the machining method, MR_i is the manufacturing resource and cutting parameter; TAD refers to tool approach direction, which means tool movement mode, and $A_i(x, y, z)$ are the coordinates of tool feed position, which can be obtained by BFS.

The entire machining element set of a part can be expressed as

$$ME = \{me_1, me_2, me_3, \dots, me_n\} \quad (2)$$

As the optimization variables of ME are not numerical values, but mathematical arrangements, their constraints cannot be expressed by explicit mathematical modeling. It is difficult to obtain the optimal tool path using conventional optimization rules and algorithms.^{21,22} GAs do not directly use numerical values as operands, but express variables in the form of genetic code and then optimize them.²³ Through genetics, crossovers, and mutations, it is possible to search for the optimal sequence combination of the worldwide genetic code, which offers the advantage of high parallelism and strong robustness.²⁴ Therefore, it is more appropriate to use a GA to solve the optimization problem of a tool path. However, due to the large amount of computation involved in traditional GAs, it is easier to prematurely converge to form a local optimal solution. In the engineering application process, this is often combined with the features of optimized objects, with improved GAs (multi-population GA, single parent GA, elite retention, etc.) being used to solve these practical problems.²⁵⁻²⁸

The principle idea of the elite retention strategy is that elite individuals who appear during stages of the evolution process of the group are no longer paired with other individuals and are directly copied to the next generation. At the same time, in order to ensure that the size of the group does not change, if elite individuals are added to the next-generation group, the least-adapted individuals in the group will be eliminated, while the “survival of the fittest” among genetic individuals will be achieved. In this genetic process, we try to ensure that elite individuals reproduce in the next generation of populations. Following the natural law of survival of the fittest, this can effectively avoid the loss of optimal individuals during the next generation, resulting in the GA being unable to converge to the global optimal solution. Therefore, we use GA based on elite retention strategy to optimize the tool path. The basic flow of the algorithm is shown in Figure 7.

Initial population construction method

The iterative computation of GAs is based on the population $P(t)$ (t stands for genetic algebra) consisting of chromosome set. The initial population size is generally represented by N . The value of N is based on actual experience and test guidance and typically ranges from 20 to 100.

Machining time calculation based on BFS

For robot machining, the machining time typically includes cutting time and machining assistance time. Among them, the machining-aided time refers to the tool change

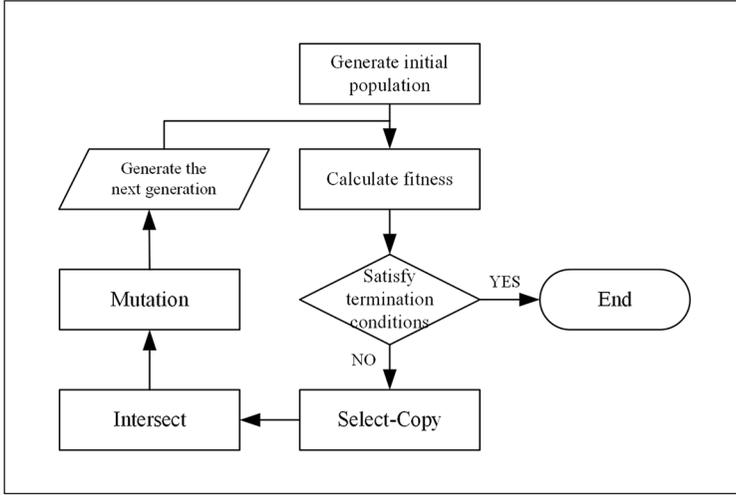


Figure 7. Flow chart of genetic algorithm.

time, the tool idle stroke time, and the workpiece transposition time. For a single cavity's tool path, the cutting time is fixed. Therefore, when the optimization objective function is established, only the machining assistance time is needed. Based on the aforementioned analysis, an objective function for optimization is established to shorten the machining assistance time

$$f(x) : \min T = \sum_{i=1}^n t_i \quad (3)$$

where T is the assistant machining time, n is the number of work steps, and t_i is the i th cavity assistant machining time.

The formula of t_i is

$$t_i = (P_{i+1} - P_i)t_{1i} + (Q_{i+1} - Q_i)t_{2i} + t_{3i} \quad (i = 1, 2, \dots, n) \quad (4)$$

where n is the number of cavities, t_{1i} is the workpiece transposition time, t_{2i} is the tool change time, t_{3i} is the tool idle stroke time, P_i is the machining orientation code corresponding to the i th cavity tool path, and Q_i is the tool's code corresponding to the i th cavity tool path.

When the machining positions P_i of two consecutive cavities are the same, the workpiece does not need to be transposed, therefore, t_{1i} is zero. When the machining positions are different, the workpiece transposition time needs to be recalculated. When two consecutive cavities use the same tool, the tool does not need to be replaced, then the tool change time t_{2i} is zero. When using different tools, the tool change time needs to be calculated. Therefore, formula (4) satisfies the conditions as follows

$$P_{i+1} - P_i = \begin{cases} 1; P_i \neq P_{i+1} \\ 0; P_i = P_{i+1} \end{cases}$$

$$Q_{i+1} - Q_i = \begin{cases} 1; Q_i \neq Q_{i+1} \\ 0; Q_i = Q_{i+1} \end{cases}$$

In formula (4), the t_{3i} is directly related to the distance between different cavities, which can be calculated by $A_i(x, y, z)$

$$t_{3i} = \frac{\sqrt{A_{i+1}^2 - A_i^2}}{v_f} \quad (5)$$

where $A_i(x, y, z)$ is the tool feed position of i th cavity, which can be obtained by BFS. v_f is the tool speed of idle stroke.

Determination of fitness function

The fitness function is used to measure the individual's merit in GAs. The fitness size determines whether a genetic individual breeds or dies. The greater the individual's fitness, the better it is. Usually, the fitness function is obtained by converting the objective function, both of which should have positive values and the same extreme points. Based on this principle, the fitness function is expressed in conjunction with formula (3) as

$$F(x) = \frac{1}{f(x)} \quad (6)$$

Selection and copy operations

The elite individuals selected in conjunction with the elite retention strategies do not cross-operate with other individuals and are directly added to the next generation through replication. At the same time, individuals with the lowest fitness are eliminated from the next generation and the number of groups remains unchanged. Elite individuals generally use fitness as the measurement standard. The number of elite retention in each generation is controlled by the replication probability P_r . The range of P_r is generally set to 0.1–0.2.

Crossover operator

Crossover is the main method for generating new individuals in GAs (the principle is similar to hybridization in the field of biology). The number of chromosomes that perform crossover operations is usually controlled by the crossover probability P_c , while the range of P_c values is generally set to 0.5–0.8. Crossover methodologies include single-point intersections, two-point crossings, and circular crossings. In this article, considering the rationality of the initial work steps sequence and to avoid destroying the potential high-quality gene fragments, a single-point crossover algorithm is used, as shown in Figure 8. The steps taken are as follows:

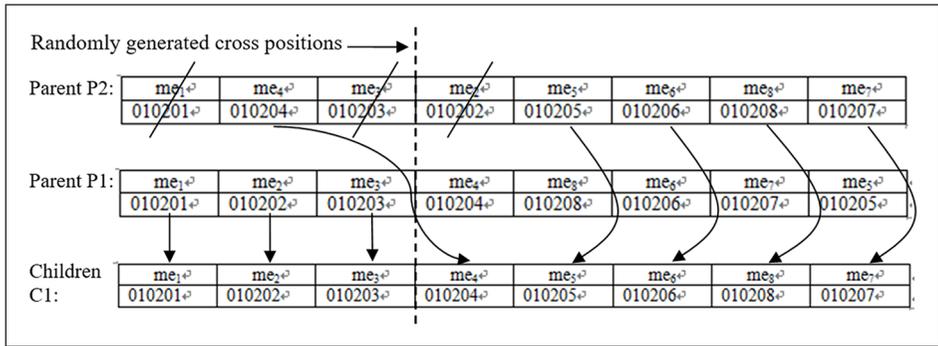


Figure 8. Single cross generation of offspring chromosomes.

Step 1. According to the method shown in Figure 8, P1 and P2 are arranged as the parent chromosomes for performing the cross operation.

Step 2. An intersection is randomly generated, dividing P1 and P2 into two segments: left gene segment and right gene segment.

Step 3. Copy the left part of the parent P1 intersection to the individual C1 of the offspring to form the left half of the C1 gene fragment.

Step 4. Delete the same gene in the parent P2 as that in the parent P1. The remaining genes are copied to the child C1 in the order of P2, forming the right half of C1, and generating the complete child C1.

Mutation operator

The number of chromosomes that produce mutations is typically controlled by the mutation probability Pm . The range of Pm values is generally set to 0.01–0.1. The conversion operation steps are as follows:

Step 1. According to the method shown in Figure 8, a reasonable initial work step sequence P1 is arranged as the chromosome for performing the mutation operation.

Step 2. Assume that the chromosome length is n , and randomly generate two natural numbers as mutation points in the interval $[1, n]$.

Step 3. Exchange the two sets of genes at the mutation site to generate a new chromosome P2.

Step 4. According to the work step constraint rule, P2 is checked for validity. If it is valid, P2 is used as a descendant group for subsequent calculation; if it is invalid, return to Step 2.

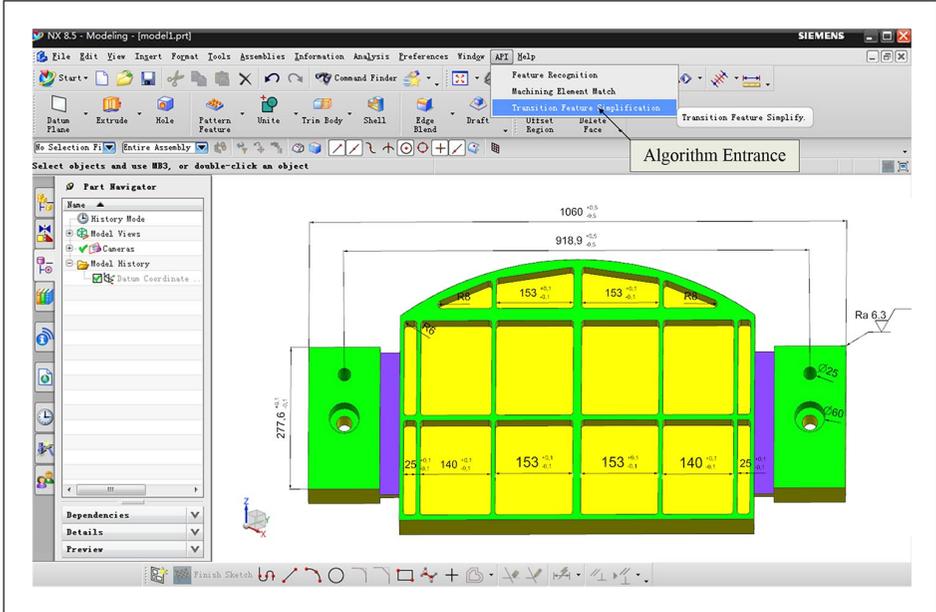


Figure 9. Multi-cavity model of an aircraft structural part.

Termination conditions

There are two termination conditions commonly used in GAs: (1) set the number of iterations N . When the number of iterations of the algorithm reaches the specified value, the operation terminates; and (2) when several iterations are performed and the optimal individual's fitness does not change significantly, the operation terminates.

Case study

To verify the effectiveness of the proposed method described in this article, the robot tool path planning of an aircraft structural part (shown in Figure 9) is used as an example. The processing of this part mainly involves the milling of 16 cavities ($Ra=6.3$); each cavity corner adopts fillet transition. In this case, the machining accuracy is not high, so a robot or five-axis machine tool milling can be used. Considering the deformation during processing, a robot with a large load and rigidity is required.

The method described in section "Algorithm modeling of BFS" is used to simplify the blend features and efficiently obtain the tool feed positions of different cavities. We make use of the application programming interface that NX provides and carry on the secondary development to NX8.5, which realizes the BFS algorithm, as shown in Figure 9.

According to the number of cavities in the part, the number of corresponding machining elements is 16. According to the method described in section "Robot tool path optimization using BFS," machining elements represent a parent entity in the

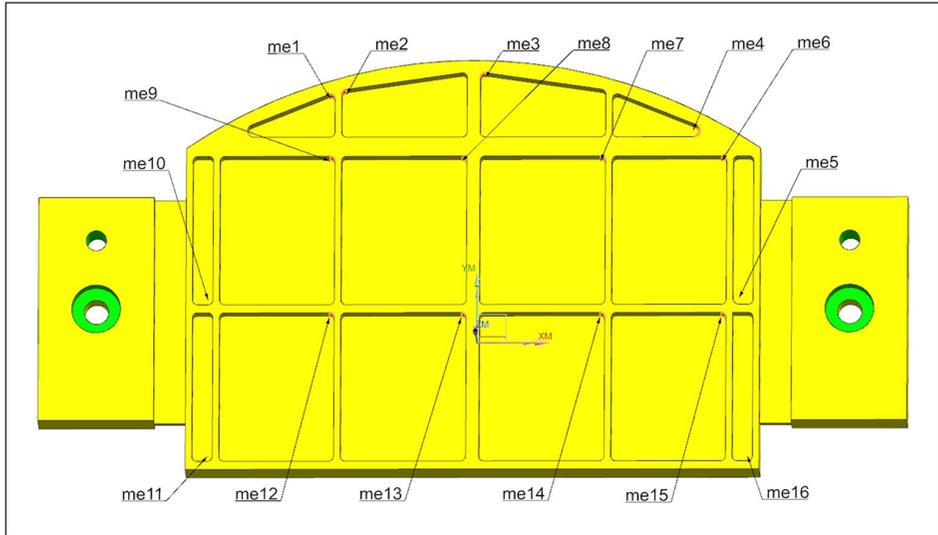


Figure 10. Machining element location (tool feed position).

initial population. Combining the blend features of the cavities and BFS algorithm, we obtain the tool feed position of 16 machining elements, as shown in Figure 10.

Combined with the 3D model as shown in Figure 10, the information of the generated parent entity P1 is shown in Table 2.

As a means for achieving the comparison verification, we tested three different tool path simulation motions. The first tool path followed cavity centroid, shown in Figure 11(a). Figure 11(b) shows the second tool path, which is zigged with contour. Figure 11(c) shows the trochoidal tool path, which follows cavity center. Figure 11(d) shows the tool path simulation motions, based on BFS.

For the tool path planning, built on BFS, if the calculation is based on the mathematical total alignment method, theoretically, the maximum evolution algebra (possible number of step sequences) is $M=16!$ In order to reduce the amount of calculations required, we set three different maximum evolution algebras to $M=100$, $M=50$, and $M=35$ times as the termination condition of the algorithm. When the number of iterations reached the maximum evolution algebra, the algorithm was automatically terminated. We used MATLAB to iteratively calculate the sample, comparing the results of the conventional GA with the improved-GA, as shown in Table 3. Table 4 shows the machining time required to use these three different tool paths.

Upon comparing the computed results of the conventional GA with the improved-GA, we are able to obtain the following results:

- In this experiment, when the terminate conditions (maximum evolution algebras) are set to $M=50$, $M=35$ times, we cannot obtain the optimal results in the maximum evolution algebras by using the conventional GA. On the contrary,

Table 2. Initial population chromosome.

ME	Coordinate (tool feed position)			Tool	n (r/min)	f (mm/min)	a_p (mm)
	x	y	z				
me_1	-180	293	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_2	-158	299	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_3	10	321	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_4	263	253	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_5	320	50	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_6	170	50	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_7	10	50	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_8	-160	50	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_9	-310	50	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{10}	-340	50	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{11}	-340	-145	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{12}	-310	-145	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{13}	-160	-145	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{14}	10	-145	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{15}	170	-145	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1
me_{16}	320	-145	50	End milling $\Phi 12 \times 30 \times 60$	900	250	1

the convergence rate of improved GA is faster and the convergence algebra is much smaller than the largest evolutionary algebra. Through analysis, we can see that the improved-GA algorithm, used in this article, has high computational efficiency.

- Since the parent P1 selected in this calculation has a good genetic gene, theoretically, according to the elite retention principle, the convergence algebra should be within $(16-16 \times 50\%)! = 40,320$ generations. The calculation results show that the convergence algebra is far less than the theoretical value, which proves the superiority of the proposed GA algorithm.
- The tool path method, based on BFS, can locate the optimal solution within the specified maximum evolutionary algebra to achieve the goal of optimizing the tool path. By combining Table 4, it can be calculated that the machining time of BFS is 10.08% faster than following cavity centroid, 19.38% faster than zig with contour, and 11.7% faster than trochoidal path.

Conclusion and future work

In this article, we have proposed and validated a robot tool path planning method, based on BFS. Some of the main contributions of this research are listed here.

- The machining element is proposed to describe the work step decisions. Considering the influence of the blend feature on the position of the tool in-feed,

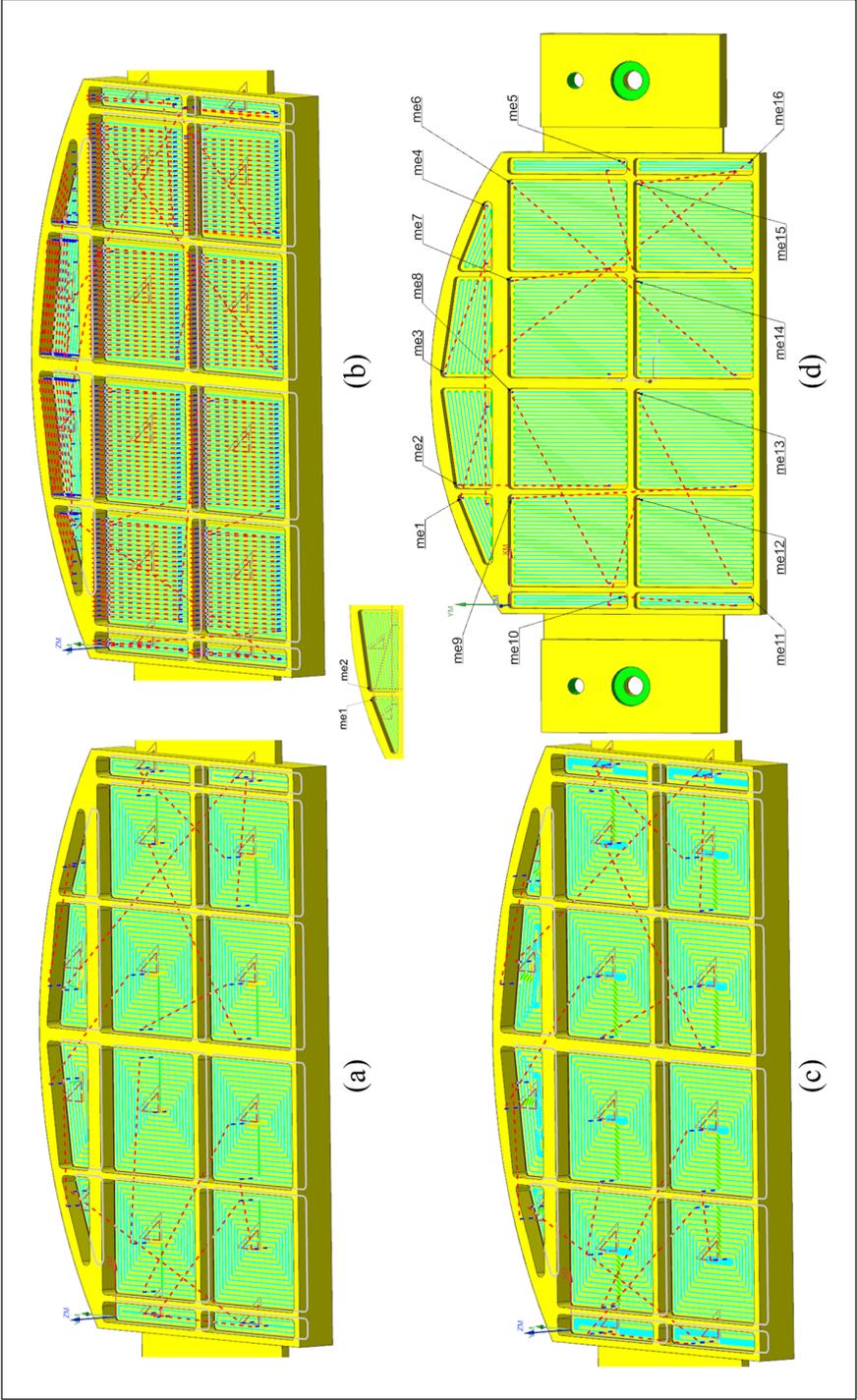


Figure 11. Tool path simulation: (a) follow cavity centroid, (b) zig with contour, (c) follow cavity centroid with trochoidal, and (d) tool path planning based on BFS.

Table 3. Comparison of conventional GA and improved GA.

N	P _c	P _m	Conventional GA		Improved GA		Optimal individual (shown in Figure 11(c))
			P _r	Convergence algebra	P _r	Convergence algebra	
100	0.7	0.05	0	87	0.2	46	me ₁₁ → me ₁₀ → me ₁₂ → me ₁₃ → me ₉ → me ₈ → me ₂ → me ₁ → me ₄ → me ₃ → me ₁₆ → me ₁₅ → me ₅ → me ₁₄ → me ₇ → me ₆
50	0.6	0.05	0	50	0.15	33	me ₁₁ → me ₁₀ → me ₁₂ → me ₁₃ → me ₉ → me ₈ → me ₂ → me ₁ → me ₄ → me ₃ → me ₁₆ → me ₁₅ → me ₅ → me ₁₄ → me ₇ → me ₆
35	0.5	0.01	0	35	0.1	25	me ₁₁ → me ₁₀ → me ₁₂ → me ₁₃ → me ₉ → me ₈ → me ₂ → me ₁ → me ₄ → me ₃ → me ₁₆ → me ₁₅ → me ₅ → me ₁₄ → me ₇ → me ₆

GA: genetic algorithm.

Table 4. Comparison of three tool path results.

Tool path type	Machining time (h/min/s)
Follow cavity centroid	02:55:47
Zig with contour	03:16:04
Trochoidal	02:59:11
BFS	02:38:04

BFS: blend feature simplification.

we designed a BFS algorithm to accurately obtain the tool feed position. On this basis, the spatial distance between different cavities can be calculated automatically. This makes tool path planning for cavity machining more convenient.

- An improved-GA is used to find the optimal tool path. The results show that compared with other three common methods the approach proposed in this article can shorten more than 10% machining time.

As blend features may exist at different corners of a cavity, as demonstrated in the example used in this article, manual intervention is often required to select the blend features to be simplified. The tool path planning is ultimately a combination of artificial experience and artificial intelligence algorithms. Although this approach reduces the intelligence of the decision-making process, it greatly reduces the computational complexity of the algorithm and improves the decision-making efficiency and enhances the practicality of the method.

The algorithm is equally applicable to five-axis machining machines. However, for the multi-cavity machining programming with blend features, as described in this article, the robot machining planning has more advantages than traditional numerical control

(NC) programming. First, the offline programming software for the robot can automatically identify and search the CAD model for point, line, and surface information to generate tracks. By combining the BFS algorithm described in this article, the tool feed position can be more accurately and quickly obtained. Second, the trajectory is associated with the CAD model features; if the model moves or deforms, the trajectory will change automatically.

In order to further improve the intelligence of the decision-making process, follow-up research should consider improvement of the algorithm. Future work is also planned to extend and improve the prototype system by intelligently screening the feed position of blend features.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by the Scientific and Technological Innovation Projects in Shanxi Universities (Grant No. 201802027), the Key Laboratory of Road Construction Technology and Equipment of MOE (Chang'an University; Grant Nos 300102259514 and 310825171004), the Natural Science Basic Research Project of Shaanxi Province, China (Grant No. 2019JM-073), the China Postdoctoral Science Foundation (Grant No. 2018M633439), and the National Natural Science Foundation of China (Grant Nos 51705353 and 51805043).

ORCID iD

Gangfeng Wang  <https://orcid.org/0000-0002-4607-6188>

References

1. Zivanovic S, Slavkovic N and Milutinovic D. An approach for applying STEP-NC in robot machining. *Robot Com-Int Manuf* 2018; 49: 361–373.
2. Wu K, Krewet C and Kuhlenkötter B. Dynamic performance of industrial robot in corner path with CNC controller. *Robot Com-Int Manuf* 2018; 54: 156–161.
3. Milosevic M, Lukic D, Antic A, et al. e-CAPP: a distributed collaborative system for internet-based process planning. *J Manuf Syst* 2017; 42: 210–223.
4. Rai JK and Xirouchakis P. Finite element method based machining simulation environment for analyzing part errors induced during milling of thin-walled components. *Int J Mach Tools Manuf* 2008; 48: 629–643.
5. Liu X, Li Y and Gao J. A multi-perspective dynamic feature concept in adaptive NC machining of complex freeform surfaces. *Int J Adv Manuf Tech* 2016; 82: 1259–1268.
6. Niknam SA, Davoodi B, Davim JP, et al. Mechanical deburring and edge-finishing processes for aluminum parts—a review. *Int J Adv Manuf Tech* 2018; 95: 1101–1125.
7. Groba C, Sartal A and Vázquez XH. Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: an application to fish aggregating devices. *Comput Oper Res* 2015; 56: 22–32.
8. Abdullah H, Ramli R and Wahab DA. Tool path length optimisation of contour parallel milling based on modified ant colony optimisation. *Int J Adv Manuf Tech* 2017; 92: 1263–1276.

9. Chen L, Pu J and Wang X. A general model for machinable features and its application to machinability evaluation of mechanical parts. *Comput Aided Design* 2002; 34: 239–249.
10. De Giovanni L and Pezzella F. An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. *Eur J Oper Res* 2010; 200: 395–408.
11. Hu L, Gu Z-Q, Huang J, et al. Research and realization of optimum route planning in vehicle navigation systems based on a hybrid genetic algorithm. *Proc IMechE Part D: J Automobile Engineering* 2008; 222: 757–763.
12. Xu X, Wang LH and Newman ST. Computer—aided process planning—a critical review of recent developments and future trends. *Int J Comp Integ M* 2011; 24: 1–31.
13. Li L, Liu F, Chen B, et al. Multi-objective optimization of cutting parameters in sculptured parts machining based on neural network. *J Intell Manuf* 2015; 26: 891–898.
14. Chu C-H, Lee C-T, Tien K-W, et al. Efficient tool path planning for 5-axis flank milling of ruled surfaces using ant colony system algorithms. *Int J Prod Res* 2011; 49: 1557–1574.
15. Kuo C-L, Chu C-H, Li Y, et al. Electromagnetism-like algorithms for optimized tool path planning in 5-axis flank machining. *Comput Ind Eng* 2015; 84: 70–78.
16. Plakhotnik D and Lauwers B. Graph-based optimization of five-axis machine tool movements by varying tool orientation. *Int J Adv Manuf Tech* 2014; 74: 307–318.
17. Dong M, Shao X and Xiong S. Flexible optimization decision for product design agility with embedded real options. *Proc IMechE Part B: J Engineering Manufacture* 2011; 225: 1431–1446.
18. Mocellin K and Cerutti X. Numerical prediction of distortions during machining of large aluminium aeronautical parts. *Mat Sci Eng Technol* 2016; 47: 699–709.
19. Qiu Y and Zhou X. Machining features recognition from triangular meshes in reverse engineering. *J Comput-Aided Des Comput Graphics* 2010; 22: 711–716.
20. Ridwan F, Xu X and Liu GY. A framework for machining optimisation based on STEP-NC. *J Intell Manuf* 2012; 23: 423–441.
21. Liu X, Hong Y, Ni ZH, et al. Process planning optimization of hole-making operations using ant colony algorithm. *Int J Adv Manuf Tech* 2013; 69: 753–769.
22. Liu XJ, Yi H and Ni ZH. Application of ant colony optimization algorithm in process planning optimization. *J Intell Manuf* 2013; 24: 1–13.
23. Abu Qudeiri JE. Production simulator system for flexible routing optimization in flexible manufacturing systems. *Proc IMechE Part B: J Engineering Manufacture* 2017; 231: 1237–1247.
24. Ouyang HB and Shen B. STEP-NC oriented process planning optimization based on hybrid genetic algorithm. *Comput Integr Manuf Syst* 2012; 18: 66–75.
25. Bo ZW, Hua LZ and Yu ZG. Optimization of process route by Genetic Algorithms. *Robot Com-Int Manuf* 2006; 22: 180–188.
26. Karimi D and Nategh MJ. Contour maps for developing optimal toolpath and workpiece setup in hexapod machine tools by considering the kinematics nonlinearity. *Proc IMechE Part B: J Engineering Manufacture* 2016; 230: 1572–1583.
27. Wang XJ, Song Z and Jia XJ. Milling-turning processing optimization for aircraft engine casing. *Comput Integr Manuf Syst* 2011; 17: 1460–1465.
28. Lee CKH. A review of applications of genetic algorithms in operations management. *Eng Appl Artif Intel* 2018; 76: 1–12.

Author biographies

Yupeng Xin received the Ph.D. degree in mechanical engineering from Northwestern Polytechnical University, Xi'an, China, in 2016. He is currently a Postdoctoral Fellow and Lecturer with the

College of Mechanical and Vehicle Engineering, Taiyuan University of Technology, Taiyuan. His research interests include smart manufacturing process planning, CPS, IoT, and digital twin. He has authored/coauthored more than 20 scientific papers in peer-reviewed journals. Currently, his research activity is mainly focused on digital twin-based smart process planning.

Shengqiang Yang is Full Professor of Mechanical Engineering at the College of Mechanical and Vehicle Engineering, Taiyuan University of Technology. He is in charge of Shanxi Provincial Key Laboratory of Precision Machining. He has authored/coauthored more than 150 scientific papers in peer-reviewed journals.

Gangfeng Wang received the Ph.D. degree in mechanical engineering from Northwestern Polytechnical University, Xi'an, China, in 2015. He is currently a Postdoctoral Fellow and Senior Engineer with the School of Construction Machinery, Chang'an University, Xi'an. His research interests include smart manufacturing informatics, knowledge-driven innovation design, and product development of intelligent construction machinery. He has authored/coauthored more than 20 scientific papers in peer-reviewed journals, including the *International Journal of Production Research*, *Applied Sciences-Basel*, *Mathematical Problems in Engineering*, and the *Science Progress*.

Richard Evans received the Ph.D. degree in enterprise social software from the University of Greenwich, London, U.K., in 2013. He is a Senior Lecturer in Human Factors for Design at Brunel University London, London, U.K. He has authored/co-authored numerous papers in peer-reviewed journals, including *Robotics and Computer Integrated Manufacturing*, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Sustainability*, and the *International Journal of Production Research*.

Fengfeng Wu received the Ph.D. degree in mechanical engineering from Beihang University, Beijing, China, in 2015. She is currently a Postdoctoral Fellow and Lecturer with the College of Mechanical and Vehicle Engineering, Taiyuan University of Technology, Taiyuan. Her research interests include aircraft assembly and precision machining.