IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 June 2023, accepted 5 September 2023, date of publication 11 September 2023, date of current version 15 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3313636

RESEARCH ARTICLE

Optimization Cutting Trajectory for Ring Knife Sponge Cutting via S-Shaped Sequencing Algorithm Based on Region Segmentation

KUN XIE AND ZHISONG ZHU
School of Mechanical Engineering, Nantong University, Chongchuan, Nantong, Jiangsu 226019, China
Corresponding author: Zhisong Zhu (zhu.zhs@ntu.edu.cn)

This work was supported by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX22_3339.

ABSTRACT Continuous cutting of sponge samples is a key technique for efficient cutting with sponge ring cutters. However, existing methods are not suitable for complex-shaped and large quantities of sponge samples. To address this issue, this paper proposes a S-shaped sequencing algorithm based on region segmentation. Firstly, the algorithm separates the sample set into multiple molecular regions using the inflexion point samples. Then, global and local sorting are performed on each molecular region to determine the optimized search sequence. Next, entry and exit points are defined, and pre-generated processing paths for the samples within the region are created. Path collisions are identified and eliminated, resulting in the optimal cutting sequence for the samples within the molecular region. Experimental results demonstrate that this algorithm effectively reduces the empty travel distance and collision occurrences during the cutting process. Compared to manual methods, this algorithm can reduce the empty travel distance by 14-25%. In the next step, the algorithm will generate processing code to facilitate the practical application of the algorithm.

INDEX TERMS Continuous cutting, molecular region, region segmentation, the machining path, collision detection.

I. INTRODUCTION

Under the background of mass production of sponge samples, the key to improving processing efficiency and quality is to reasonably select the tool cutting path [1], [2]. During the cutting process, the workpiece processing time comprises two parts: the tool cutting time and the tool idle time. The contour length of the sponge sample in the sponge sample drawing is fixed, so the time to process the contour is fixed, that is, the tool cutting time is fixed. Therefore, the length of the workpiece processing time depends on the idle time of the tool, which is the idle travel time, and the length of the idle travel is related to the sample sorting [3], [4], [5], [6], [7]. In the sponge cutting system, the cutting order of sponge samples can be manually sorted on commercial CAD/CAM software based on the operating experience of the operator, or set automatically on the industrial computer according to different sorting algorithms. Compared with manual sorting, automatic sorting is more accurate and efficient [8], [9].

Unlike discontinuous cutting, which uses algorithms such as simulated annealing [10], genetic [11], cuckoo [12] and ant [13], continuous cutting requires the tool to cut all contours at once during the cutting process. Regarding the processing of continuous cutting, many scholars have studied it. Liu [14] proposed the concept of a common cutting seam and the use of common edge cutting in practical production, thus achieving a continuous cut. A mathematical model for laser cutting path planning when regular and non-regular parts are lined up with common edges, using common edge cutting, was developed by Liu et al. [15] based on the theory of graphology to form a continuous cutting method. Lv [16] proposed a graph theory-based ''one-stroke'' and a co-edge cutting method for array rectangles. Xu's research [17] has solved the problem of processing irregularly shaped parts in co-edge cutting and further optimised the continuous cutting method for co-edge cutting.

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei.

The continuous cutting approach taken in the above literature is more restrictive and unsuitable for complex machining environments. This paper proposes an S-shaped sorting algorithm based on region segmentation, which gradually divides the sample layout to be cut into multiple regions, and sorts and searches the samples in each region to finally establish the cutting order of the samples and the tool cutting path.

II. PROBLEM DESCRIPTION AND FORMATION

During the sponge processing, the servo motor operates the XY-C axis to precisely cut the sponge sample. The Y axis moves the ring knife, the X axis moves the workpiece, and the C axis adjusts the ring knife's deflection, enabling seamless cutting of the sponge material. During the entire cutting process, the ring knife does not break away from the sponge material, and the cut sponge sample does not break away from the material. Currently, most discontinuous cutting methods select a contour point on the cutting object as the entry and exit point, which is not possible with sponge ring knives. To address the issue of continuous cutting in the sponge ring cutting machine, the algorithm used in this paper independently selects the entry and exit points. Collision handling is another critical issue in protecting the integrity of the cut samples. Processing routes formed between samples can collide with the peripheral samples (transition lines passing through other sample boundaries or interior areas), thereby damaging the samples.

III. S-TYPE SORTING ALGORITHM FLOW BASED ON REGION SEGMENTATION

During the cutting process with a ring-shaped tool, there are instances where samples are cut through or undergo repeated cutting, which does not meet the process requirements. To avoid this situation, the processing sample area can be divided. The cutting direction is divided into two types: upward and downward. The area sample starts the search from the bottom sample, called upward cutting; the area sample starts the search from the top sample, called downward cutting. Firstly, according to the leftmost inflexion point sample in the sample sequence, divide the upper part of the sample into a region, which is cut upward, sort and process path planning for the samples in this region. The cutting direction is changed to down, then according to the leftmost inflexion point sample in the remaining sample sequence, divide the lower part of the sample into a region, sort and process path planning for the samples in this region. Repeat this process until all the samples have been searched, presenting an S-shaped cutting path. During the search of samples, collision detection and handling are required. The process flow of the S-type sorting algorithm based on region segmentation is shown in Fig. 1.

IV. REGIONAL SEGMENTATION

Unlike the cutting samples with multiple and scattered contour points, the centroid is the unique geometric center, designated as a feature point of the sample. It reflects the position of the sample within the set and serves as the sorting criterion. An initial region, *PR*(Primary Region), is expanded by extending the corner point samples. The inflexion point samples are unsearched and are divided into two types: top leftmost and bottom leftmost dailies, denoted as *LS*. To find *LS* from a collection of unsearched samples, proceed as follows:

Step 1: The set of unexplored samples P;

Step 2: Find the sample A with the smallest value of the x-coordinate of the centre of the set;

Step 3: Add the eligible samples from the sample set to the empty set according to the rule f_{ls};

Step 4: Let the number of samples in the set S be t. If $t = 0$, then select A as LS; if $t = 1$, select the sample in S as LS; neither, Perform step 2 for S.

The rules f_{ls} used for the above process are shown below.

$$f_{ls} = \begin{cases} x_{bl} < x_{ar} \\ y_{bc} < y_{ac}, & \text{Cutting upwards} \\ y_{bc} > y_{ac}, & \text{Cutting down} \end{cases} \tag{1}$$

In the above equation, x_{bl} is the x-coordinate value of the lower left vertex of the envelope rectangle of sample B to be added, x_{ar} is the x-coordinate of the upper right vertex of the envelope rectangle of A, y_{bc} and y_{ac} are the y-coordinates of the shape centres of B and A respectively. It can be seen that the left boundary of the enveloping rectangle of B needs to lie to the left of the centroid of A. When the cut is up, the centroid of B has to lie below the centroid of A to be added to the set S. When the cut is down, the centroid of B has to lie above the centroid of to be added to S.

Based on *LS*, the unsearched samples that meet the criteria are added to the *PR*. As shown in Fig. 2, the *CS* (candidate sample) will only be added to the *PR* if either the left boundary of the bounding rectangle of the *CS* is to the left of the centroid of *LS*, or the centroid of the *CS* is to the left of the right boundary of the bounding rectangle of *LS*.

The *PR* needs to expand by including the surrounding samples that have smaller volumes but have not been placed into any region yet, resulting in the formation of a *MR* (molecular region). The rule of the expansion is to find the sample with the rightmost right boundary of the envelope rectangle of the current region, and expand the samples with the center of gravity located on the left side of the right boundary of the envelope rectangle of the sample to the region.

V. REGIONAL SORTING

A. OVERALL SORTING

Initially, the preliminary processing sequence of the *MR* is formed based on the Y-coordinate values of the centroid of the samples, either in descending order or ascending order. If the search is carried out in this order, there may be a confusion in the machining route, resulting in the tool cutting through the machining sample. Therefore, the *MR* is divided into multiple parallel regions. If the centre of the right-hand

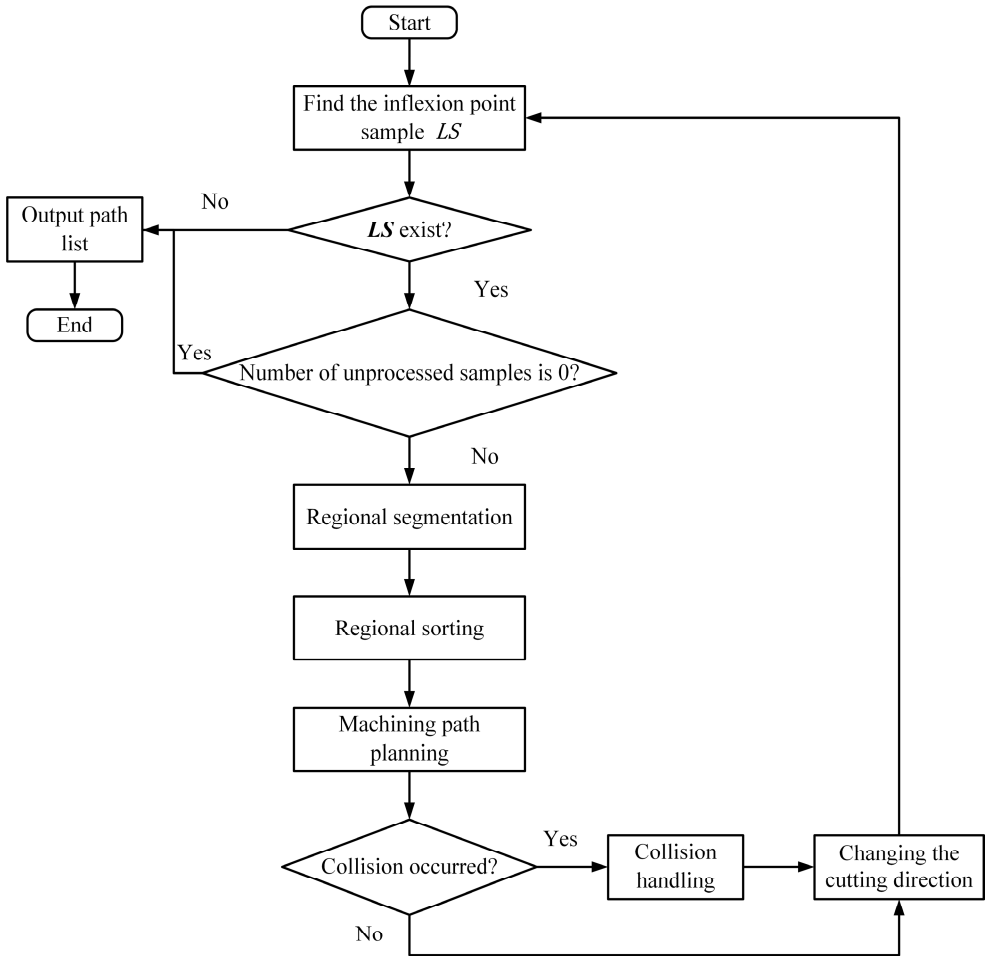

FIGURE 1. Algorithm flow.

sample lies between the upper and lower boundaries of the enveloping rectangle of the leftmost sample, the former is said to be the right same-row sample of the latter, and the latter is the left same-row sample of the former. The area formed by same-row samples is referred to as the same-row region. When there are no same-row samples existing, create a same-row region independently. Adjusting the processing sequence of samples within the same-row region can optimize the processing route to the maximum extent.

The following description will detail the process of forming same-row regions. A sample A is known and the same-row sample of the sample A has four cases: (i) there are left and right same-row samples; (ii) there are only right same-row samples; (iii) there are only left same-row samples; (iv) there are no same-row samples. In both the first and second cases, sample A has a left-hand same-row sample(LSS), and it is necessary to first find the LSS of A, and then add the right same-row sample of A to a set based on the LSS. As shown in Fig. 3, sample A first finds its left counterparts B and C according to the definition of same-row samples. Since the centroid of C is on the far left, C is the leftmost same-row sample. According to the definition of same-row samples, it can be known that D and E are both same-row samples of C, and finally put five samples in the same collection. In the third case, sample A has only right same-row samples, at which point A is the LSS of the right-hand sample, and the same-row samples of A are added to a set according to the same-row sample definition. In the fourth case, Sample A is not processed or modified in any way.

The same-row area can also be divided into multiple same-column areas. A same-column sample is a sample whose centroid is located between the left and right boundaries of another sample's enveloping rectangle. The same-column area is the area formed by same-column samples. When a sample does not have any samples in the same column, it forms a separate column region on its own. The search rule for the MR is consistent with the cutting direction, where the search is done upwards, and the areas are searched from bottom to top according to their heights. Conversely, the search is done downwards, from the bottom to the top, for the opposite direction. The search rule for the same-row area is to search for each area from left to right, and the search order for the samples in the same-column area is consistent with the region cutting direction.

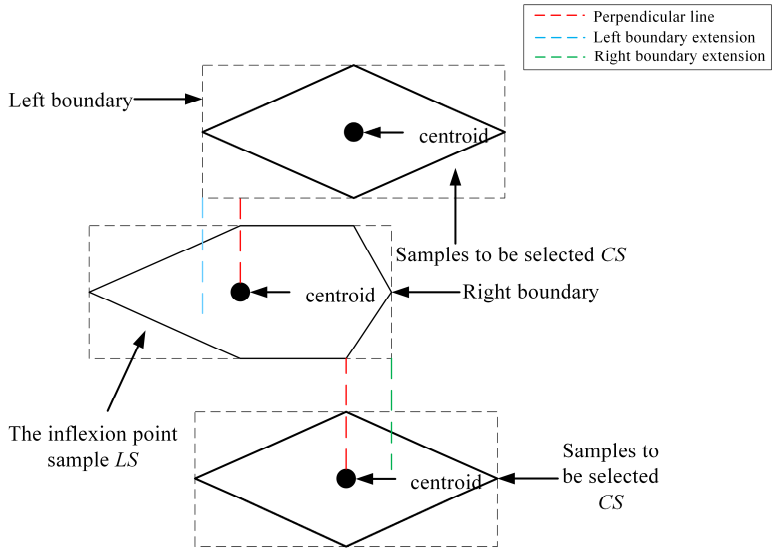

FIGURE 2. Prototype meets the conditions for expansion.

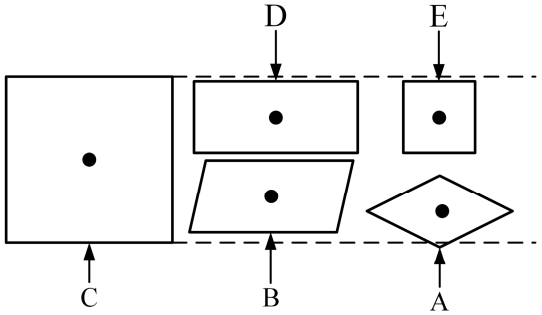

FIGURE 3. The process of the right sample finding the same-row sample.

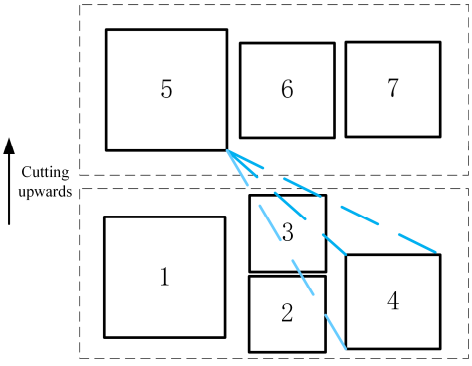

FIGURE 4. Collision occurs in sequential search samples.

As shown in Fig. 4, when searching for samples according to the search rule for the same-row area, collisions may occur when the last sample in the same-row area and the next sample to be searched form a transition line. To minimize collisions, it is necessary to use a backtracking method to reverse the search order of the same-row area. To clarify, for the multiple same-row regions within *MR*, the search order for samples in the last region, denoted as r_n, remains unchanged, r_{n-1} changes the search order according to the relative position of r_n, and so on until the first region r_1. The search order of the ith searched region r_i of *MR* is known, the shape center of the first searched sample r_{i1} in r_i is denoted *LB*. The shape center of the first searched sample $r_{(i-1)1}$ in r_{i-1} is denoted LA_{-1} and the shape center of the last searched sample $r_{(i-1)-1}$ is denoted LA_{-1}. The midpoint of the line connecting LA_1 and LA_{-1} is denoted *mid* and the r_{i-1} inverted search order needs to satisfy f_1.

$$f_1 : x_1 < x_2 \tag{2}$$

where x_1 and x_2 are the coordinate values of *LB* and *mid* respectively. f_1 means that when r_{i1} is close to $r_{(i-1)1}$, the element order of r_{i-1} needs to be reversed, so that the sample r_{i1} is searched next to $r_{(i-1)1}$ to reduce collisions.

Due to the special position of r_1, if r_1 can be reversed according to the backtracking method, f_2 must be satisfied in order to reverse the search order; otherwise, it will remain unchanged.

$$f_2 = \begin{cases} y_{a1} > y_{l2}, & cd = 0 \\ y_{a2} < y_{l1}, & cd = 1 \end{cases} \tag{3}$$

where cd represents the cutting direction, $cd = 1$ means cutting up; $cd = 0$ means cutting down. y_{a1} and y_{a2} represent the y-coordinate values of the lower left and upper left vertices of the envelope rectangle of the inflexion point sample respectively, y_{l1} and y_{l2} represent the y-coordinate values of the lower left and upper left vertices of the enveloping rectangle of the leftmost element r_{1l} of r_1, respectively. f_2 indicates that there are only two cases where a can be inverted in order, (i) The cutting direction is upward, and the upper boundary of the inflexion point piece is located below the

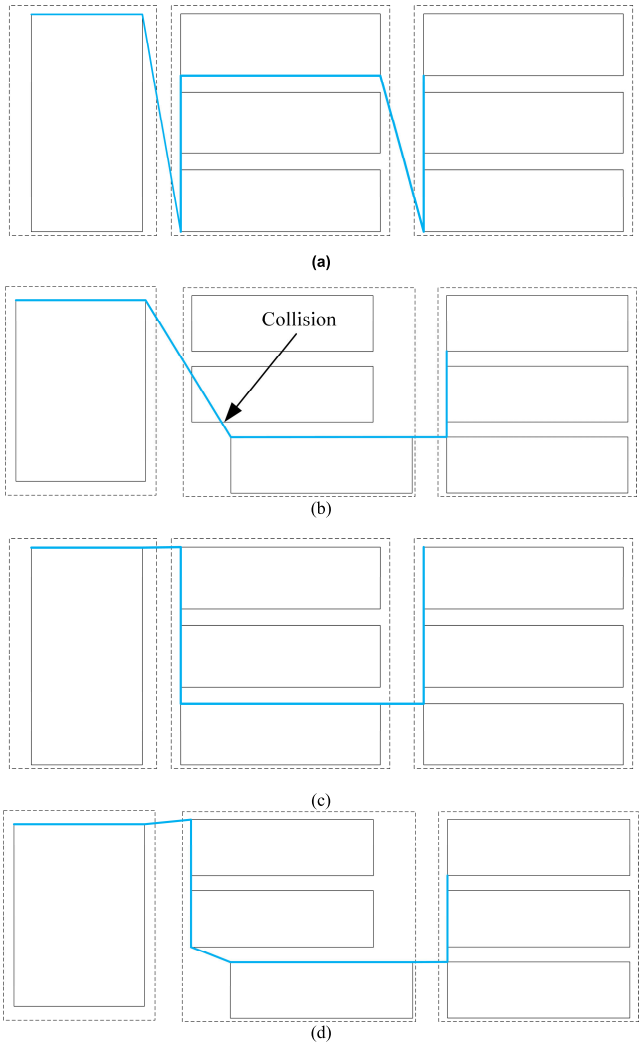

FIGURE 5. (a) The N-shaped route leads to an increase in the length of the cutting path. (b) The N-shaped route leads to collisions. (c) The S-shaped route reduces the length of the path. (d) The S-shaped route eliminates collisions.

lower boundary of r_{1l}; (ii) The cutting direction is downward, and the lower boundary of the inflexion point piece is located above the upper boundary of r_{1l}.

B. PARTIAL SORT

As shown in Fig. 5a-b, after the overall sorting, the samples within the same-row regions are searched according to the search rules of the same-column region, which increases the length of the empty travel and even leads to collisions. The algorithm solves the aforementioned problem by adjusting the search order of the same-row regions through local sorting, resulting in a processing path that exhibits an S shape. Invert the search order of the middle same-column area shapes, as shown in Fig. 5c, and create an S-shaped cutting path, as illustrated in Fig. 5d, to eliminate collisions from Fig. 5b. The search order of the first search region M_1 and the last search region M_{-1} in the *SRA*(same row area) remains

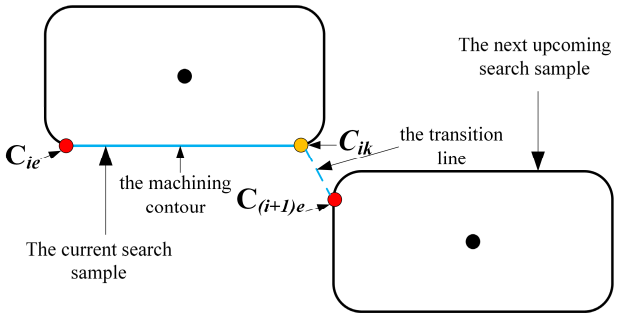

FIGURE 6. Diagram of the machining route.

unchanged. Due to the limitations of regional expansion, the range of the number of elements n in the *SRA* is [1, 5]. When i is an even number, invert the search order of M_i. When i is an odd number, do not change the search order.

VI. MACHINING PATH PLANNING

A. CHOICE OF CUT-IN AND CUT-OUT POINTS

The sorted samples within the molecular region need to be searched and processed to generate machining paths. The machining path is the transition route between two machining samples and consists of two parts: the machining profile and the transition line. The machining profile is the contour of the sample between the incoming and outgoing points of the current search sample, with the contour of the sample being either clockwise or anti-clockwise. The transition line is the line formed between the cut-out point of the current search sample and the cut-in point of the next upcoming search sample. As shown in Fig. 6, the machining path needs to establish the entry point C_{ie} and the exit point C_{ik} for the current search sample C_i and the entry point $C_{(i+1)\mathrm{e}}$ for the next upcoming search sample. When C_i is the first sample searched, its feed point C_{ie} is selected as the first point in the set of contour points; when C_i is not the first sample searched, only C_{ik} and $C_{(i+1)\mathrm{e}}$ need to be selected as the feed point C_{ie} is determined when searching the previous sample of C_i. There are mainly four sources of point pair sets for the C_{ik} of the exit point and the $C_{(i+1)\mathrm{e}}$ of the entry point:

a. Point-to-point. This means selecting one vertex on C_i and one vertex on C_{i+1}.

b. Front point to back edge. That is, choose a vertex V_{ip} from C_i and choose the edge $E_{(i+1)}l$ $(l = 1, 2, \ldots, n-1)$ of C_{i+1} in turn, and find the nearest point $V_{(i+1)q}$ from the point to the edge $E_{(i+1)l}$. If $V_{(i+1)q}$ is a point (other than an endpoint) on $E_{(i+1)l}$ that meets the requirement.

c. Back point to front edge. That is, nearest point (excluding endpoint) from C_{i+1}'s vertex to C_i's edge.

d. The closest point. The closest points between C_i an C_{i+1}.

The selection of the shortest machining path is also the selection of C_{ik} and $C_{(i+1)\mathrm{e}}$. The selection process requires constraints, which are as follows.

$$\begin{cases} l\left(C_{ik}, C_{(i+1)\mathrm{e}}\right) \cap C_i = C_{ik} \\ l\left(C_{ik}, C_{(i+1)\mathrm{e}}\right) \cap C_{i+1} = C_{(i+1)\mathrm{e}} \end{cases} \tag{4}$$

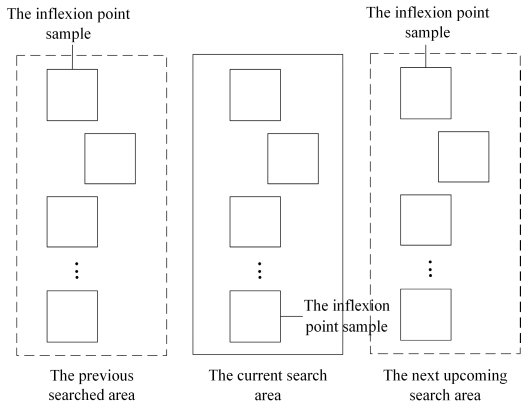

FIGURE 7. Collision area.

$$l\left(C_{ik}, C_{(i+1)}\mathrm{e}\right) \cap C_n = \emptyset, \{n \neq i, n \neq i+1 \,|n \in \{1, 2, \ldots, m\}\} \tag{5}$$

$$\begin{cases} S = \min\left(S\left(C_{ie}, C_{(i+1)}\mathrm{e}\right)\right) \\ S\left(C_{ie}, C_{(i+1)}\mathrm{e}\right) = S\left(C_{ie}, C_{ik}\right) + S\left(C_{ik}, C_(i+1)\mathrm{e}\right) \\ S\left(C_{ie}, C_{ik}\right) = \min\left\{S\left(C_{ie}, C_{ik}\right)_C, S\left(C_{ie}, C_{ik}\right)AC\right\} \end{cases} \tag{6}$$

where $l\left(C_{ik}, C_{(i+1)}\mathrm{e}\right)$ is the transition line formed by points C_{ik} and $C_{(i+1)}\mathrm{e}$, and $S(\mathrm{C}_{ie}, \mathrm{C}_{ik})_{\mathrm{C}}$ and $S(\mathrm{C}_{ie}, \mathrm{C}_{ik})\mathrm{AC}$ are the lengths of the contour of the cut-in point clockwise and counterclockwise to the cut-out point respectively. Equation (4) defines that $l\left(C_{ik}, C_{(i+1)}\mathrm{e}\right)$ does not generate any other intersections with C_i and C_{i+1}. Equation (5) defines that $l\left(C_{ik}, C_{(i+1)}\mathrm{e}\right)$ collides with other samples outside of C_i and C_{i+1}, and it prefers collision-free transition lines. Equation (6) ensures the shortest path. Equation (5) and (6) ensure the selection of the shortest path with collision-free transition lines.

B. COLLISION HANDLING

To avoid cutting through the samples when a collision occurs between the transition line and a sample, it is necessary to repeat the cutting on the collided sample, which increases the idle travel distance. Therefore, measures must be taken to detect and avoid such collisions. In order to reduce the detection time, it is necessary to limit the collision detection area. As shown in Fig. 7, the collision areas for the current search area are the previous search area, the current search area and the upcoming search area. If the current search area is the first area, there is no previous search area in the collision area; if the current search area is the last area, there is no upcoming search area in the collision area. If the machining path formed by the samples within the current search area collides with the samples in the upcoming search area, a portion of the samples from the upcoming search area is inserted into the current search area. At the same time, the upcoming search area is regenerated.

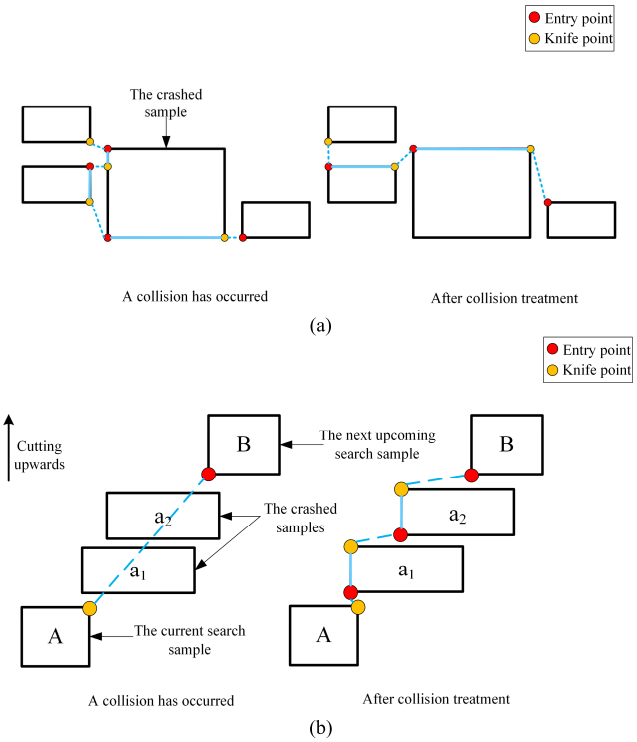

FIGURE 8. Collision handling. (a) When the collided sample is already searched and within the cur-rent search region, the previously generated machining path is eliminated. (b) When cut-ting upwards, the algorithm searches for the bottom-leftmost sample at each step.

The pre-generated transition lines may collide with both unsearched shapes and already searched shapes. Assume that $Q = \{q_1, q_2, \ldots, q_m\}$ is the set of searched samples, $P = \{p_1, p_2, \ldots, p_n\}$ is the set of unsearched samples, the current search area is R, the current searched sample is q_m ($q_m \in R$), the set of processing paths is $T = \{t_1, t_2, \ldots, t_{m-1}\}$, the next sample to be searched is p_1, and the collision is a single sample q_w. The samples are processed accordingly according to the type of q_w:

a. Unsearched sample

q_w is added to R, P deletes q_w, q_m generates a machining path with q_w, then generates a machining path with p_1, and the resulting machining path is added to T.

b. Searched sample

(I) If q_w is in the set R, as shown in Fig. 8a, the two samples adjacent to q_w in Q generate a processing path and place q_w at the last position of Q. q_m generates a machining path with q_w, q_w then generates a machining path with p_1, and the resulting machining path is added to T. (II) If q_w is not in set R, it is added to Q. q_m and q_w generate a machining path, and q_w then generates a machining path with p_1. The generated machining path is added to T. As q_w is a searched sample, the tool does not need to cut the complete profile of q_w added to Q at this point in the machining process, only the generated machining profile.

When colliding with multiple shapes using pre-generated transition lines, select the search order of multiple shapes

TABLE 1. Results of simulation experiments.

Number of samples	Optimised sorting	Number of collisions	Average empty travel/mm	Empty travel/mm	Time/s	Optimisation rate/%
58	*No*	2	10650	8134	604	23.6
58	*Yes*	1	10650	8021	544	24.0
63	*No*	2	11560	9656	601	16.5
63	*Yes*	1	11560	9132	580	21.0
68	*No*	2	12193	10373	671	14.9
68	*Yes*	1	12193	10295	626	15.6
73	*No*	5	13225	11341	740	14.2
73	*Yes*	4	13225	11032	616	16.6
78	*No*	6	14089	12010	803	14.7
78	*Yes*	4	14089	11461	714	18.6

in the set based on the cutting direction. Cut up, selecting to process the lower leftmost sample at a time; cut down, selecting to process the upper leftmost sample at a time. As shown in Fig. 8b, when cutting upwards, the current search sample A first forms a machining path with the leftmost lower sample a_1, after which a_1 then forms a machining path with a_2, and finally a_2 then forms a machining path with B.

VII. EXPERIMENTAL VERIFICATION

The algorithm takes into account important constraints in actual machining on the machine tool, such as avoiding collisions between samples and minimizing empty travel distance. This ensures that the generated processing paths have practical engineering significance. In the future, the algorithm can generate processing G-code for actual production purposes. The algorithm test in this paper was developed in a Python environment on a PC (specifications: Windows 10 operating system, Intel Core i5, 2.5HZ processor, 16GB RAM). Multiple complex sample sets were selected to validate the algorithm.

A. REGION EXPANSION STRATEGY

The size of the *MR* will have an impact on the length of the generated empty paths. If the regions are too small, there will be an increased number of collisions between the pre-generated transition lines and the shapes from other regions when searching the shapes in that region. On the other hand, if the regions are too large, there will be an increased number of collisions between the pre-generated transition lines and samples from the surrounding area of that region. *PR* is further expanded to obtain the optimal sample region. The region expansion strategy employed for the set of 68 samples involves expanding the *PR* for 0 to 5 iterations. The resulting lengths of the empty travel distances and the number of collisions after expansion are depicted in Fig. 9. It can be observed that when the expansion count for *PR* is 0, the pre-generated transition lines experience the highest number of collisions and result in the longest path. On the other hand, when the expansion count is 2, the number of collisions is minimized, and the path is also the shortest.

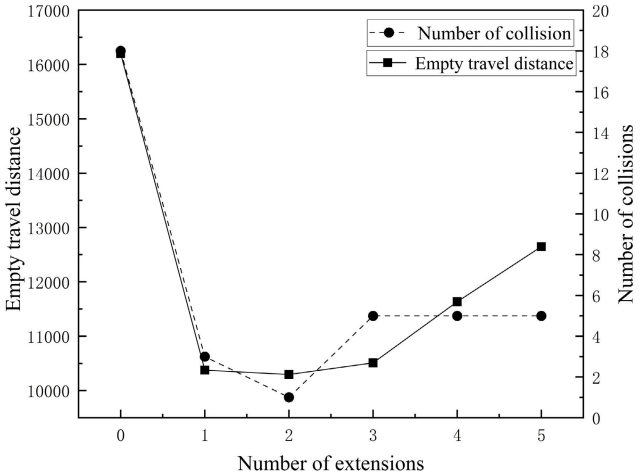

FIGURE 9. Empty trip distances and collision counts obtained for different expansion iterations.

B. COLLISION AVOIDANCE TECHNIQUE

Five sets of complex sample collections were selected, with sample sizes of 58, 63, 68, 73, and 78, for comparison before and after applying collision avoidance techniques. When a collided previously searched sample is in the same region as the sample that generates the transition line, collisions between the transition line and the previously searched sample are eliminated by adjusting the sequence of subsequent samples. The planning results are shown in Table 1, indicating a significant reduction in the number of collisions between the generated transition lines and surrounding samples compared to not using collision avoidance techniques. As the number of samples to be cut increases, the number of collisions also increases.

In Fig. 10, the red dots represent the entry points of the samples, the yellow dots represent the exit points of the samples, and the blue line segments represent the empty

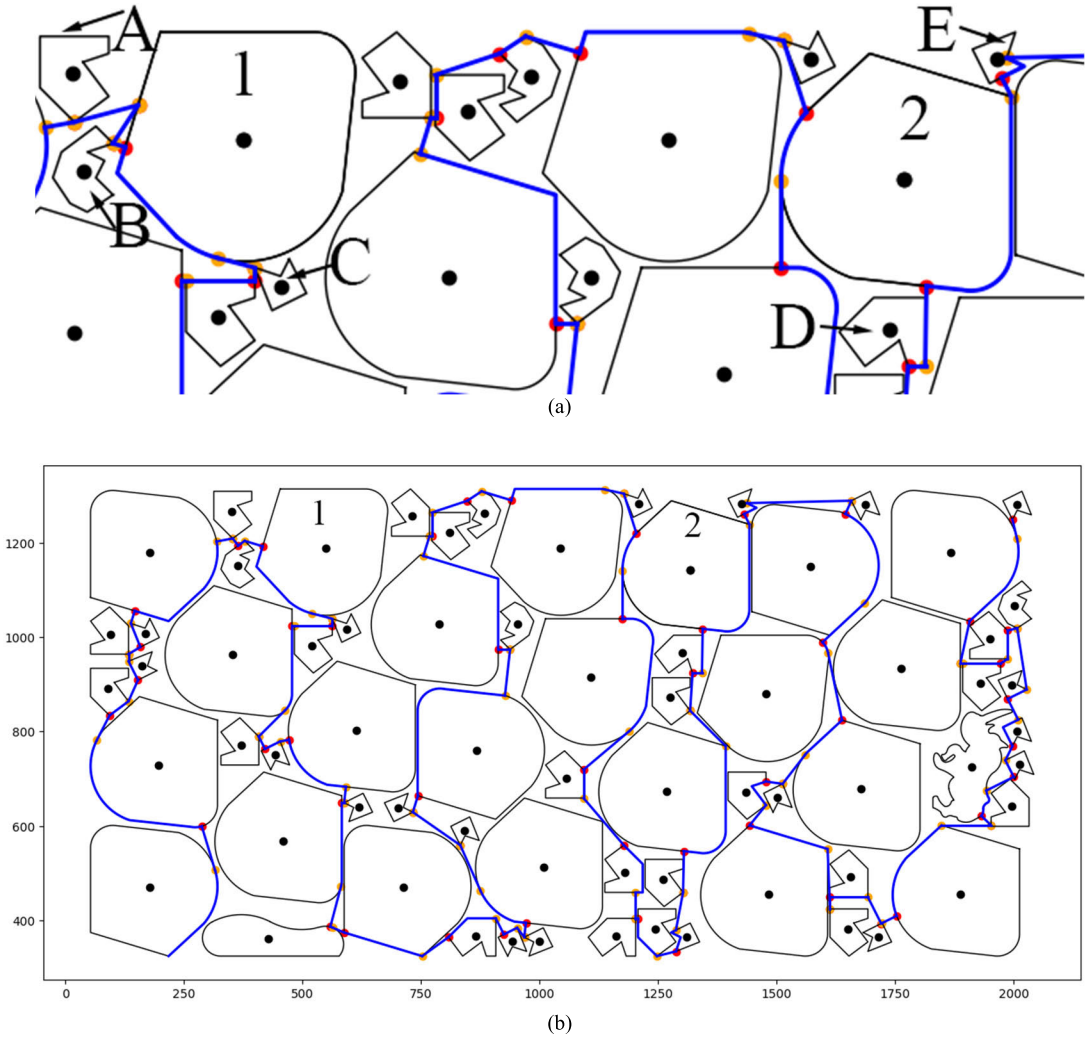

FIGURE 10. Simulation Experimental Results: (a) The local schematic diagram before collision avoidance. (b) The collision avoidance schematic diagram.

travel segments. From Fig. 10(a), it can be observed that sample *B* and sample 1 are in the same molecular region, and when generating the machining paths for both, sample 1 has already been searched. After collision avoidance, as shown in Fig. 10(b), the machining paths involving sample *A* and 1, as well as the path between sample 1 and sample *B*, are eliminated to eliminate collisions. Although sample 2 has been searched, it cannot be collision-avoided because it is not in the same region as the neighboring samples *D* and *E*. As a result, the tool needs to cut repeated paths on the contour of sample 2. Compared to the average empty travel distance in manual sorting, the experimental range of reduction achieved through collision avoidance is between 14% and 25%. This significant improvement demonstrates the effectiveness of collision avoidance.

VIII. CONCLUSION

To address the challenge of continuous cutting of a large number of complex sponge samples, this paper proposes an S-shaped sorting algorithm based on region segmentation. This method defines the entry and exit points of the samples, enabling efficient continuous cutting of sponge samples. The algorithm begins by globally sorting the samples within the molecular regions. It then introduces the concept of the same-row sample to determine the initial search order of samples within the same-row region. By combining the concept of the same-column region, further local sorting of samples within the same-row region is performed to optimize the search order. The algorithm determines collision detection regions to reduce collision detection time and standardizes the handling of transition lines colliding with single and multiple samples, thus avoiding damage to the integrity of the samples during the machining process. Compared to discontinuous cutting methods, the most significant feature of the method proposed in this paper is the achievement of continuous machining paths, which means there are no tool lifts and no damage during cutting. Compared to algorithms that do not utilize collision avoidance techniques, this method can

significantly reduce the sorting time of samples, minimize collision occurrences, and decrease the idle travel distance. In the next step, this algorithm will generate machining G-codes and be applied to actual production on machine tools.

REFERENCES

[1] A. A. Petunin and C. Stylios, "Optimization models of tool path problem for CNC sheet metal cutting machines," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 23–28, 2016.

[2] S. C. Park and B. K. Choi, "Free-form die-cavity pocketing," *Int. J. Adv. Manuf. Technol.*, vol. 22, nos. 11–12, pp. 864–872, Dec. 2003.

[3] G. Zhou, C. Zhang, F. Lu, and J. Zhang, "Integrated optimization of cutting parameters and tool path for cavity milling considering carbon emissions," *J. Cleaner Prod.*, vol. 250, Mar. 2020, Art. no. 119454.

[4] E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propag.*, vol. AP-15, no. 5, pp. 692–693, Sep. 1967.

[5] A. G. Chentsov, P. A. Chentsov, A. A. Petunin, and A. N. Sesekin, "Model of megalopolises in the tool path optimisation for CNC plate cutting machines," *Int. J. Prod. Res.*, vol. 56, no. 14, pp. 4819–4830, Jul. 2018.

[6] R. Dewil, P. Vansteenwegen, and D. Cattrysse, "A review of cutting path algorithms for laser cutters," *Int. J. Adv. Manuf. Technol.*, vol. 87, nos. 5–8, pp. 1865–1884, Nov. 2016.

[7] C. Zhang, F. Han, and W. Zhang, "A cutting sequence optimization method based on Tabu search algorithm for complex parts machining," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 233, no. 3, pp. 745–755, Feb. 2019.

[8] M. Hajad, V. Tangwarodomnukun, C. Jaturanonda, and C. Dumkum, "Laser cutting path optimization using simulated annealing with an adaptive large neighborhood search," *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 1–4, pp. 781–792, Jul. 2019.

[9] K. Kiani, M. Sharifi, and M. Shakeri, "Optimization of cutting trajectory to improve manufacturing time in computer numerical control machine using ant colony algorithm," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 228, no. 7, pp. 811–816, Jul. 2014.

[10] G.-C. Han and S.-J. Na, "Global torch path generation for 2-D laser cutting process using simulated annealing," *Intell. Autom. Soft Comput.*, vol. 4, no. 2, pp. 97–108, Jan. 1998.

[11] V. A. Kandasamy and S. Udhayakumar, "Effective location of micro joints and generation of tool path using heuristic and genetic approach for cutting sheet metal parts," *Int. J. Mater. Forming*, vol. 13, no. 2, pp. 317–329, Mar. 2020.

[12] Z. Zhang and J. Yang, "A discrete cuckoo search algorithm for traveling salesman problem and its application in cutting path optimization," *Comput. Ind. Eng.*, vol. 169, Jul. 2022, Art. no. 108157.

[13] N. Hatem, Y. Yusof, A. Z. A. Kadir, K. Latif, and M. A. Mohammed, "A novel integrating between tool path optimization using an ACO algorithm and interpreter for open architecture CNC system," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 114988.

[14] J. L. Liu, "The application on common gap of numerical control cutting machine," *J. Huaibei Vocational Tech. College*, no. 1, pp. 83–84, 2004, doi: 10.3321/j.issn:0258-7025.2004.10.026.

[15] H. X. Liu, X. Wang, M. Zhou, and L. Cai, "Nozzle path planning of edge-shared nested workpiece in laser cutting," *Chinse J. Laser*, vol. 30, no. 10, pp. 1269–1274, 2004, doi: 10.3321/j.issn:0258-7025.2004.10.026.

[16] Y. J. Lv, Q. J. Han, and Y. Q. Rao, "Path optimization in metal cutting based on common line cutting method," *Mach. Des. Manuf.*, no. 6, pp. 120–122, 2011, doi: 10.19356/j.cnki.1001-3997.2011.06.046.

[17] C. C. Xu, "Research on the common cutting technology for the special-shaped parts," *Mech. Res. Appl.*, vol. 32, no. 6, pp. 191–193, 2019, doi: 10.16576/j.cnki.1007-4414.2019.06.061.

KUN XIE was born in Hefei, Anhui, China. He is currently pursuing the master's degree with Nantong University. His current research interests include advanced manufacturing technology, machine tool processing, and intelligent robotics.

ZHISONG ZHU was born in Taizhou, Jiangsu, China. He received the master's degree in mechanical and electronic engineering from the Nanjing University of Science and Technology, in 2004. He started teaching with Nantong University, in 2004. His current research interests include advanced manufacturing modes, advanced manufacturing systems, and mechanical design and manufacturing. He was awarded the title of Senior Experimentalist, in 2008.