Problem Chosen	2020 APMCM Summary Sheet	Team Control Number
A		2020010080021

Laser Marking Hatch Contour Generation

Summary

Laser marking has been widely used in industrial processing. For different hatching methods, the operating efficiency and the clarity of the marking pattern are also different. We need to realize the zigzag parallel and contour parallel hatch of the single-layer and multi-layer contour pattern respectively based on different parameters, evaluate the algorithm, and give optimization strategies. Three models are established: Model I: Zigzag Parallel Hatch Model; Model II: Contour Parallel Hatch Model; Model III: Performance Evaluation Function Model.

For model I, firstly we calculate the number of parallel lines, then construct an intersection matrix with the intersection points of the equally spaced parallel lines and the contour curve, and sequentially shrink the intersection points into the contour line, determine the discontinuity point by the slope of the corresponding point of the intersection matrix position, finally get the transfer matrix. At last, the transfer matrix is divided into blocks, and the elements of each block matrix are connected in an alternating sequence from left to right and from right to left to obtain horizontal equidistant hatching lines. According to the model we built, the total length of the hatching lines with a horizontal line spacing of 1mm and 0.1mm in the single-layer contour pattern in Attachment 1 is 797mm and 9972mm, the number of horizontal lines of zigzag parallel hatch is 88 and 907, and the average elapsed time based on the multiple runs of hatching program is 289ms and 1886ms, the ratio of elapsed time is 6.52; the total length with a horizontal line spacing of 1mm and 0.1mm in the multi-layer contour pattern in Attachment 2 is 1011mm and 11079mm, the number of horizontal lines is 126 and 954, the average elapsed time is 266ms and 1952ms; the ratio of elapsed time is 7.34.

For model II, we built a point-based polygon filling algorithm: the contour curve with smaller curvature is directly contracted inward while using the Douglas-Puck algorithm to retain the inflection point and delete some unimportant smooth points, which effectively reduces the number of cycles; In the place where the curvature is large, the invalid loop is eliminated through the interference point. Finally, the contour termination condition is explained. According to the model, the total length of the hatching lines in the single-layer contour pattern in Attachment 1 is 402mm and 1021mm, the number of circles of contour parallel hatch is 11 and 85, and the average elapsed time is 326ms and 4090ms, the ratio is 12.55; the total length in the multi-layer contour pattern in Attachment 2 is 408mm and 1099mm, the number of circles is 12 and 106, the average elapsed time is 440ms and 5030ms; the ratio is 11.27.

In order to evaluate the efficiency of the algorithm, we constructed a performance function to evaluate the marking efficiency by analyzing the parameters of the calculation results in the above questions and the actual situation, then based on this performance function, we use genetic algorithms to propose strategies and directions for the further improvement of the model so that it can meet the efficiency requirement of actual industrial applications.Finally, the advantages and disadvantages of the model are given

Keywords: Laser Marking; Polygon Filling Algorithm; Performance Function

Contents

1 Introduction

1.1 Problem Background

Laser is one of the most important inventions of the 20^{th} century, and its application in industrial processing is becoming more and more extensive. As one of the significant applications of laser, laser marking uses the special effect of the interaction between laser and material to process the designed characters and patterns on the surface of the material. Compared with traditional mechanical engraving, chemical corrosion, ink printing, and other methods, laser marking is a non-contact, pollution-free, wear-free, and permanent marking processing technology. It has the advantages of low cost, high stability, flexibility, and reliability.

Figure 1: Laser marked image

The effect used by the laser marking machine is the "vaporization effect", that is, when a high-energy laser beam irradiates the surface of the material to remove part of the reflected light, the high energy in the laser will be absorbed by the material and converted into heat, so The temperature of the material surface rises in a short period, and when it reaches the vaporization temperature of the material, the surface of the material will be marked by instantaneous vaporization and evaporation.

1.2 Restatement of the Problem

For different materials and different process parameters, the effects of lasers are also different. Direction parallel hatch and contour parallel hatch are two basic filling methods. For the parallel direction filling method, the laser moves in parallel to the originally selected reference direction. Whereas the parallel hatch of contour lines is generated in a spiral manner along the curve keeping a constant distance from the curve boundary. Which type of hatch is actually used depends largely on the influence of the marking material and the processing process on the specific processing task. We need to build mathematical models to solve the following problems:

Question 1: Under the condition of only considering the horizontal zigzag parallel hatch, realize the zigzag parallel and contour parallel hatches of the single-layer contour pattern in Attachment 1, and realize the hatch under the two input parameters according to the graphic coordinate data of Attachment 1.

(1) Internal contraction of boundary distance 1mm, hatch line spacing 1mm;

(2) Internal contraction of boundary distance 0.1mm, hatch line spacing 0.1mm.

Under the two groups of parameters, calculate:

(a) the total length of hatching lines of the hatched curves subject of zigzag parallel and contour parallel hatch;

(b) the number of horizontal lines of zigzag parallel hatch;

(c) the number of circles of contour parallel hatch;

(d) the average elapsed time based on the multiple runs of hatching program;

(e) the ratio of elapsed time for program running under conditions of parameter groups (2) and (1).

Question 2: Realize zigzag parallel and contour parallel hatch of the mutually nested multi-layer contour patterns in Attachment 2. Parameters setting and other calculation contents are the same as Question 1.

Question 3: Check the elapsed time of hatching algorithm and analyze its performance to provide a strategy or direction for optimizing the performance and efficiency of the hatching algorithm.

1.3 Our Work

The hatch tool of laser marking machine can be used to hatch specified 2D-compound curve graph, and the setting of different hatch parameters have a great impact on the processing effects of different materials.To solve these problems and further explore marking efficiency, our team will do the following:

- State assumption and make notations. Ignoring some insignificant impacts.
- Establish two polygonal parallel hatch models:zigzag and contour parallel hatch. Then record the filling method, running time, scanning times, length and other data of the two models under different parallel scan line spacing.
- Establish a performance evaluation function to test our model, and optimize the hatch algorithm based on genetic algorithm.

2 General Assumptions

- **Assumption 1:** In the actual situation, the influence of the inflection point delay on the program running time is negligible, and it is considered that there is no deviation between the marking physical diagram and the diagram in the program design.
- **Assumption 2:** The filled equidistant contour line must be a closed curved polygon.
- **Assumption 3:** All the curves in this article are moved in parallel, and the zigzag parallel hatch only uses 0° degree scan line for filling.
- **Assumption 4:** In the actual production of laser marking, there is no accident caused by equipment failure

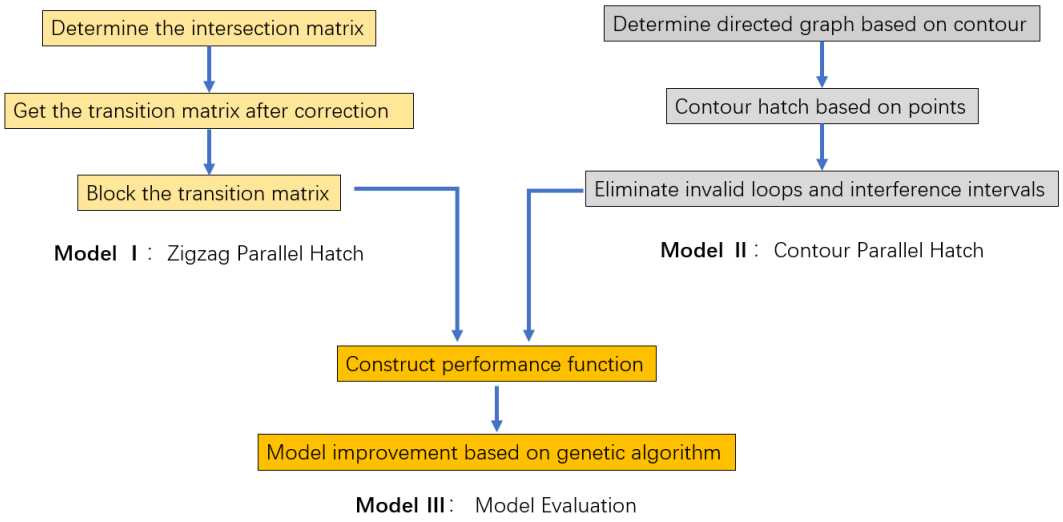


Figure 2: Model Overview

Table 1: Symbol Definition Instruction

Symbol	Description	Unit
d	The boundary distance	mm
s	The hatch line spacing	mm
Lz	The total length of hatching lines of the hatched curves subject of zigzag parallel	mm
Lc	The total length of hatching lines of the hatched curves subject of contour parallel	mm
Nz	The number of horizontal lines of zigzag parallel hatch	○
Nc	The number of circles of contour parallel hatch	○
t_p	The average elapsed time based on the multiple runs of hatching program	ms
r_{21}	the ratio of elapsed time	○
N	Intersection matrix of equidistant horizontal lines	○
N^t	Transfer matrix of equidistant horizontal lines	○
V	Vertex set	○
A	Arc set	○
G	Directed closed graph	○
ip	interference point	○
np	The new point of the i^{th} directed graph when interference occurs	○
Ii	Interference interval set	○
M	Nested multi-layer contour pattern border set	○
t_A	Actual running time of laser marking	ms
P	Performance function	○

3 Notations

4 Model I: Zigzag Parallel Hatch

4.1 Zigzag Parallel Hatch of the Single-layer Contour Pattern

For the zigzag parallel hatch method of a single-layer contour pattern, first find the minimum and maximum ordinates of each contour point in the accessory coordinate system by traversing, and then set several equal-spaced parallel lines with an inclination of 0^o degree based on the given parameters. It may be assumed that there are a total of m horizontal equal-spaced parallel lines.

$$m = [\frac{y_{max} - y_{min}}{d}] + 1 \tag{4.1}$$

When the straight line is tangent to the contour curve, we define that the number of intersections of the two straight lines is zero, so it is obvious that each horizontal line and the contour curve of a single connected domain must have an even number of intersection points, denoted as $n_i^j (j = 1, 2, \cdots, 2k)$. Store the intersection point in the matrix N.

$$\mathbf{N_{m\times 2k}} = \begin{pmatrix} n_1^1 & n_1^2 & \dots & & & \\ n_2^1 & n_2^2 & n_2^3 & n_2^4 & \dots & \\ \vdots & & & & & \\ n_i^1 & n_i^2 & \dots & \dots & n_i^{2k-1} & n_i^{2k} \\ \vdots & & & & & \\ n_m^1 & n_m^2 & \dots & & & \end{pmatrix} \tag{4.2}$$

The elements in the matrix N are points on the contour curve. In actual laser marking, the start and end points of each parallel line need to be moved according to the given parameters. We stipulate that n_1^1 is always the starting point, that is, the filling parallel line is always alternately moved from left to right and from right to left, and get a new intersection matrix $N^{'}$ by modifying the boundary distance:

$$\mathbf{N^{'}_{m\times 2k}} = \begin{pmatrix} n_1^1 + d & n_1^2 - d & \dots & & & \\ n_2^1 + d & n_2^2 - d & n_2^3 + d & n_2^4 - d & \dots & \\ \vdots & & & & & \\ n_i^1 + d & n_i^2 - d & \dots & \dots & n_i^{2k-1} + d & n_i^{2k} - d \\ \vdots & & & & & \\ n_m^1 + d & n_m^2 - d & \dots & & & \end{pmatrix}$$

Here $\pm d$ represents the distance that the abscissa of each point in the original intersection matrix moves d in the positive or negative direction of the x-axis.

When the i^{th} parallel line intersects with the contour curve to produce 2k(k>1) intersection points, it will cause the next parallel line to produce additional (2k-2) discontinuities. In order to determine the position of these discontinuities, we make a straight line parallel to the tangent direction of the original intersection point through the corresponding vertex in $N^{'}$ and intersect

the next parallel line at a certain point, which is called the transfer vertex. Fill in the corrected vertex information into the matrix to obtain the transfer matrix

$$\mathbf{N}^{\mathbf{t}}_{\mathbf{m\times 2k}} = \begin{pmatrix} n_1^1+d & n_1^2-d & \dots & & & \\ n_2^1+d & n_2^2-d & n_2^3+d & n_2^4-d & \dots & \\ n_3^1+d & n_3^2-d & n_2^{t3} & n_2^{t4} & \dots & \\ \vdots & & & & & \\ n_i^1+d & n_i^2-d & \dots & \dots & n_i^{2k-1}+d & n_i^{2k}-d \\ \vdots & & & & & \\ n_m^1+d & n_m^2-d & \dots & & & \end{pmatrix} \tag{4.3}$$

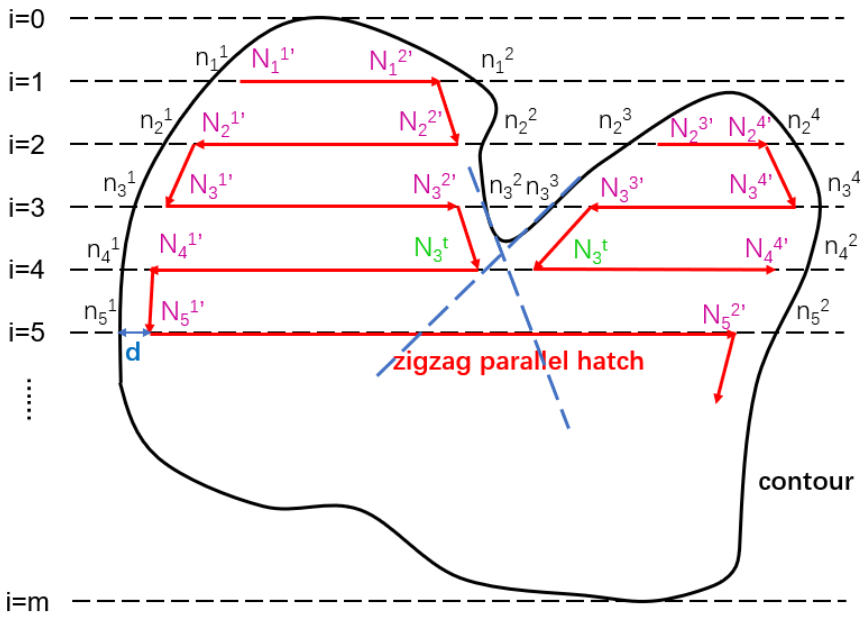


Figure 3: Zigzag parallel hatch schematic diagram

So far, we have obtained all the vertices of the zigzag parallel hatch method. We divide the transfer matrix into several m rows and 2 columns of small matrices to caculate the total length of hatching lines and the number of horizontal lines, each small matrix represents a zigzag polyline, which starts from n_1^1 and is connected alternately from left to right and from right to left.

$$N^t_{m\times 2k} = \sum_{j=1}^{k} N^t((2j-1), 2j) \tag{4.4}$$

4.2 Zigzag Parallel Hatch of the Mutually Nested Multi-layer Contour Patterns

For the parallel equidistant polyline filling in the multi-layer area, the intersection of the outer contour curve is the same as the single connected domain. The difference is that there are multiple nested contours inside, which will greatly increase the number of the elements in the

intersection matrix of the previous section , which will have a greater impact on the running time of the program.

The area filling of the multi-connected domain can be equivalent to the composite of several single connected domains and the multi-connected domain as shown in the Figure 4. Below we will analyze the multi-connected domain shown in the Figure 4.

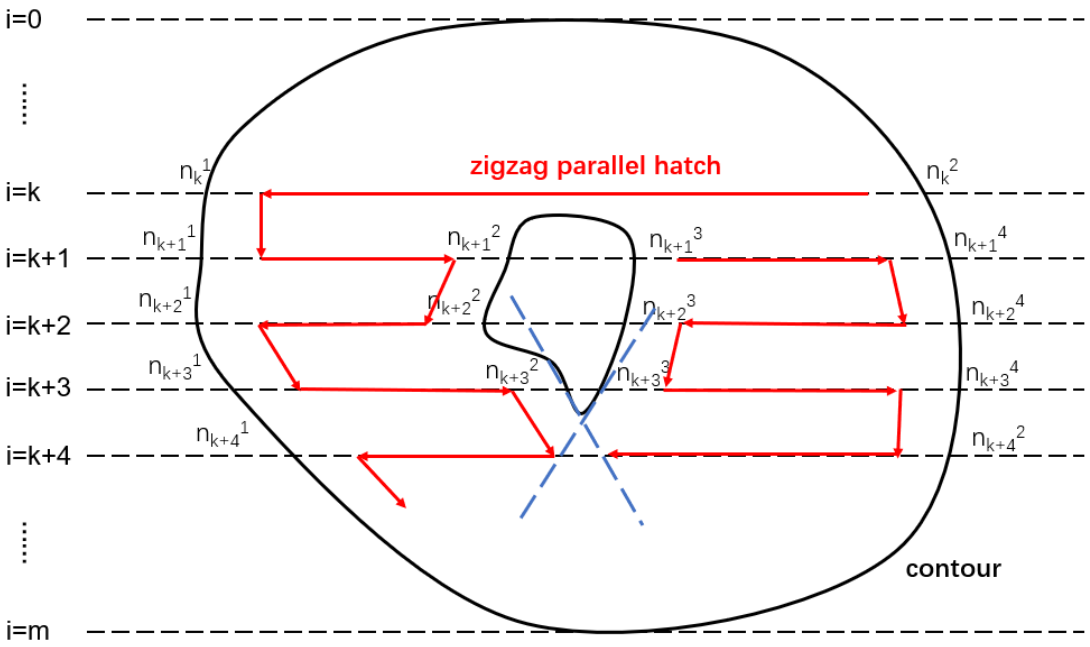


Figure 4: Zigzag parallel hatch schematic diagram

As shown in the figure above, when the number of intersections between the equidistant horizontal line and the contour curve is greater than 2, a new scan line will be generated. Here we follow the definition of the intersection matrix and transition matrix in the previous section, so we get k scan polylines. Besides, the starting point and ending point of each line segment can be obtained by the definition of the transition matrix.

4.3 Algorithm

Algorithm 1: Zigzag parallel hatch

Input: d,Attachment 1.(X_i, Y_i),Attachment 2.(X_i, Y_i)
Output: Lz,Nz,t
m based on equation (4.1);
for *i=1 to m* **do**
    Intersection matrix of equidistant horizontal lines based on equation (4.2) ;
    Transfer matrix of equidistant horizontal lines based on equation (4.3) ;
    Record the length of each lines(lz);
    Lz=Lz+lz;
end
Nz is half of the number of elements in the intersection matrix N;

5 Model II: Contour Parallel Hatch

5.1 Contour Parallel Hatch of the Single-layer Contour Pattern

First, import the data in Attachment 1 into Matlab for drawing, connect the coordinate points with straight lines, and observe that the resulting graphics are irregular polygons.

$$\begin{cases} G = G^{(i)}(V_i, A_i) \\ V_i = \{v_1^{(i)}, v_2^{(i)}, \cdots, v_n^{(i)}\} \\ A_i = \{\overrightarrow{a_1^{(i)}}, \overrightarrow{a_2^{(i)}}, \cdots, \overrightarrow{a_n^{(i)}}\} \\ \overrightarrow{a_k^{(i)}} = (v_k^{(i)}, v_{k+1}^{(i)}) \end{cases} \tag{5.1}$$

where $G^{(i)}(V_i, A_i)$ is the i^{th} contour parallel line, here ;$v_k^{(i)}$ is the k^{th} vertex of the i^{th} contour parallel line; $\overrightarrow{a_k^{(i)}}$ is the k^{th} arc of the i^{th} contour parallel line, here we call $v_k^{(i)}$ the tail of the $\overrightarrow{a_k^{(i)}}$,$v_{k+1}^{(i)}$ the head of the $\overrightarrow{a_k^{(i)}}$; we record the directed graph formed by the data points in Attachment 1 as the first contour line.

In the process of obtaining the contour parallel hatch family, each internal polygon can be regarded as each edge of the adjacent peripheral polygon is translated inward according to the given parameter distance.So we think that the relationship between the coordinates of the inner polygon and the adjacent outer polygon is:

$$G^{(i)}(V_i, A_i) = f(G^{(i-1)}(V_{i-1}, A_{i-1}), s) \quad (i > 1) \tag{5.2}$$

For the function f, we have the following description:

$$(v_{t+1}^{(i-1)}, v_{t+1}^{(i)}) = [(\overrightarrow{a_t^{(i)}} + \overrightarrow{a_{t+1}^{(i)}}) \times \vec{k}] \cdot s \tag{5.3}$$

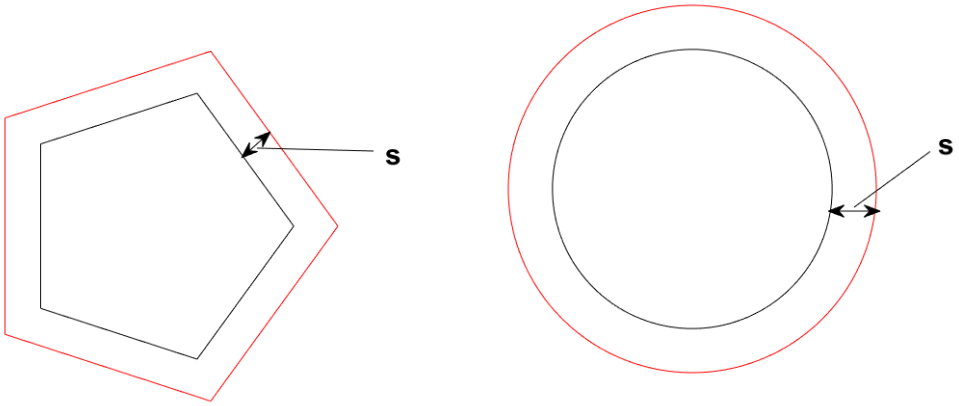
$$= [(v_t^{(i)}, v_{t+2}^{(i)}) \times \vec{k}] \cdot s \tag{5.4}$$

where $(v_{t+1}^{(i-1)}, v_{t+1}^{(i)})$ is the vector from the (t+1) vertex in the $(i-1)^{th}$ isometric contour to the (t+1) vertex in the i^{th} isometric contour;$\vec{k}$ is the unit vector facing outwards perpendicular to the two-dimensional coordinate.

Considering that the calculation efficiency of the edge offset polygon method is low, and the result is not ideal due to more invalid loops, we shrink based on points to get the contour parallel hatch curve.

For the case where the adjacent two sides of the polygon are smoother and the curvature is small, one vertex of the inner contraction curve can be determined directly from the three adjacent points of the outer curve. Consider a special case: when the figure contour is a regular polygon or a circle, the internal shrinkage contour parallel lines can be directly obtained from the given parameters.

However, when the curvature of the adjacent two sides is large, the preliminary equidistant lines generated by the point-based equidistant offset method will have intersections, so the preliminary equidistant lines are divided into two types: valid loops and invalid loops(as shown in Figure 6). The invalid loop must be deleted when the polygon is offset equidistantly.



(a) Polygon internal contraction (b) Circular internal contraction

Figure 5: Regular graphics internal contraction

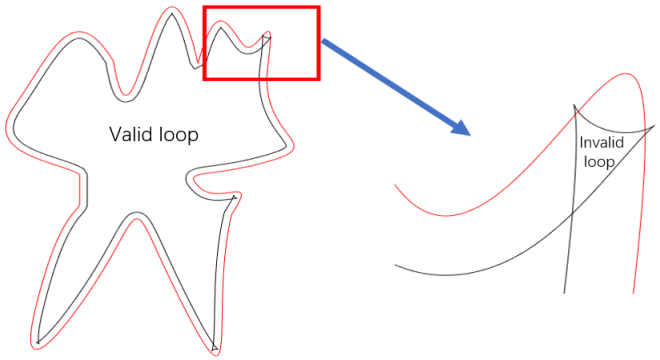


Figure 6: Invalid loop in irregular polygon

To delete the invalid loops of the preliminary equidistant line, we define the interference point and the interference interval, and compare the elements of the arc set A_i in the directed graph $G^{(i)}$ pairwise. If the intersection of $\overrightarrow{a_k^{(i)}}$ and $\overrightarrow{a_{k+m}^{(i)}}$ is not an element in the vertex set V_i(meet the following inequality), then we call this intersection point as an interference point(ip), and accordingly, call $[v_k^{(i)}, v_{k+m+1}^{(i)}]$ an interference interval.

$$\overrightarrow{a_k^{(i)}} \times \overrightarrow{a_{k+m}^{(i)}} < 0 \quad (k = 1, 2, \cdots, n, k+m = 1, 2, \cdots, n) \tag{5.5}$$

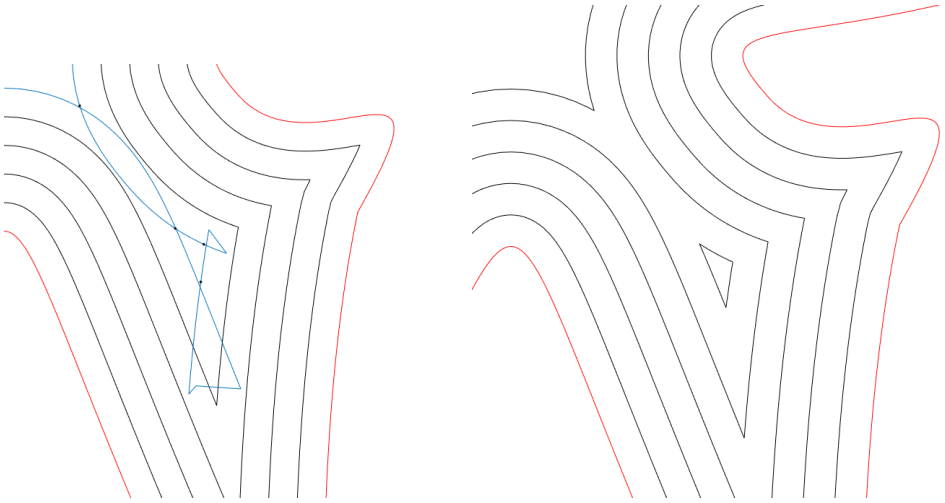
According to the definition of invalid loop and interference interval, it is known that a local invalid loop corresponds to an interference interval. Therefore, we only need to delete all vertices in the interference interval and replace this interference interval with a suitable point to eliminate the invalid loop.

For the purpose of improving the calculation efficiency, when there is only one interference point in the i^{th} directed graph, we replace the interference interval with the interference point.

$$np = ip \tag{5.6}$$

However, when there is interference as shown in the Figure 7a , we need to find several

points to replace each interference interval. In view of the fact that the goal of filling is to maximize the filling rate, in other words, to maximize the area of the union of the circles, we find the corresponding target point by controlling the area of the gap between two adjacent contour lines as small as possible.



(a) Multiple interference points in $G^{(i)}$ (b) Contour lines after eliminating invalid loops

Figure 7: Before and after processing multiple interference points

$$\begin{aligned} &\min\ S = S(G^{(i-1)}) - \sum_{q} S(G_q^{(i)}) \\ &s.t. \quad \begin{cases} G^{(i)'}(V_i, A_i) = f(G^{(i-1)}(V_{i-1}, A_{i-1}), d) \\ V_i^{'} = (V_i \bigcup np) - Ii \\ np \in Ii \end{cases} \end{aligned} \tag{5.7}$$

In summary, we get the final internal contour parallel curve as follows:

$$\begin{cases} V_i^{'} = (V_i \bigcup np) - Ii \\ A_i^{'} = \{\overrightarrow{a_1^{(i)'}}, \overrightarrow{a_2^{(i)'}}, \cdots, \overrightarrow{a_m^{(i)'}}\} \\ \overrightarrow{a_k^{(i)'}} = (v_k^{(i)'}, v_{k+1}^{(i)'}) \\ G = G^{(i)'}(V_i^{'}, A_i^{'}) \end{cases} \tag{5.8}$$

where $V_i^{'}$ is the union of each vertex of the original directed graph with the interference point after deleting the vertices in the interference interval, and $A_i^{'}$ is the set of directed arcs formed in order on the basis of the new vertex set.

The problem of how to determine the maximum value of i in the directed graph $G^{(i)}$ is a special application of the interference interval. When the equation (5.9) holds, the directed graph $G^{(p+1)}$ has a certain area outside $G^{(p)}$, at this time i reaches the maximum p.

$$[G^{(p+1)} \bigcup G^{(p)}] \neq G^{(p)} \tag{5.9}$$

5.2 Contour Parallel Hatch of the Mutually Nested Multi-layer Contour Patterns

For the filling method of contour parallel hatch of the mutually nested multi-layer contour patterns, the contraction of the contour line of the outermost contour is the same as that of the single connected domain. The definitions of the invalid ring, interference point, and interference interval are also the same as the previous section. The difference is that in in a multi-layer region, the innermost contour line obviously does not shrink to a point, but the end point is the inner contour curve(M_j), and the start point of the inner contour parallel hatch($G_j^{(1)}$) is the inner contour curve(M_{j+1}). Therefore, we redefine the directed graph in the previous section as follows:

$$\begin{cases} G_j = G_j^{(i)}(V_{ji}, A_{ji}, M_j) \\ V_{ji} = \{v_{j1}^{(i)}, v_{j2}^{(i)}, \cdots, v_{jn}^{(i)}\} \\ A_i = \{\overrightarrow{a_{j1}^{(i)}}, \overrightarrow{a_{j2}^{(i)}}, \cdots, \overrightarrow{a_{jn}^{(i)}}\} \\ \overrightarrow{a_{jk}^{(i)}} = (v_{jk}^{(i)}, v_{j(k+1)}^{(i)}) \\ M = \sum_{j}^{\alpha} M_j \end{cases} \tag{5.10}$$

where α is the maximum number of loops in a multi-layer domain,when the given contour is a single-layer contour pattern, the value of α is 1.

In the section 5.1, we used equation 5.9 as the basis to judge the end point of the innermost contour, but this is obviously not applicable in multi-layer domains, so we use equation 5.11 to judge when G_j^p reaches the end point.

$$[G_j^{(p+1)}(V, A, M_j) \bigcap M_{j+1}] \subsetneqq \bigcap M_{j+1} \tag{5.11}$$

5.3 Algorithm

To conclude, when calculating the parallel hatch curve of the contour, we can use the greedy algorithm to simplify the problem into three parts, and obtain the local optimal solution respectively, then approximate the optimal solution of the overall problem. The algorithm can be get as follows:

6 Results

6.1 Isometric fill pattern

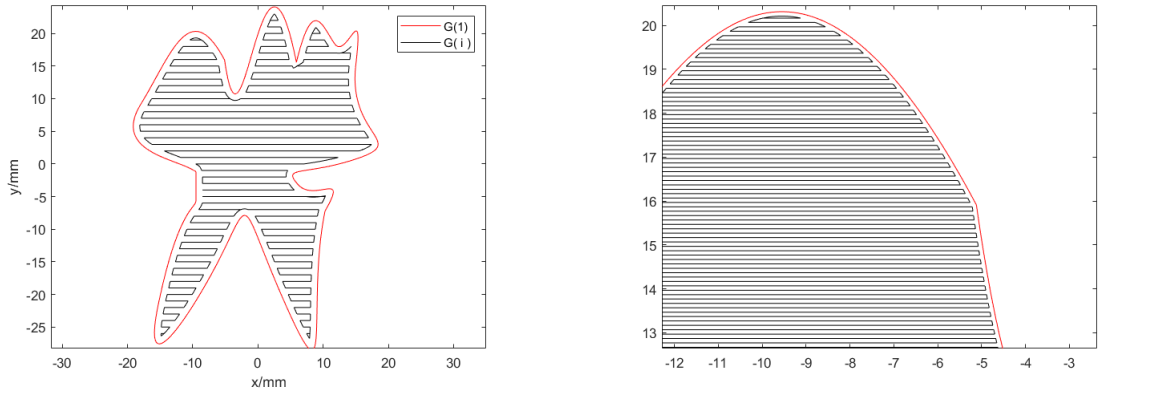
Below we give the filled curve renderings of the filling curve. For the pattern with a spacing of 0.1mm, ,we only give a partial enlarged drawing because the spacing is too small.

6.2 Calculation results

The data table of the calculation results is shown in the table 2

Algorithm 2: Contour parallel hatch

Input: s,Attachment 1.(X_i, Y_i),Attachment 2.(X_i, Y_i)
Output: Lc,Nc,t
$G_1^{(1)}$=G(Attachment 1 or 2);
for *j=1 to* α **do**
  Determine the value of p according to equation 5.9 or equation 5.11 ;
  for *i=1 to p* **do**
    The preliminary equidistant line based on equation (5.2);
    The interference point and interference interval based on equation (5.5);
    New point under interference conditions based on equation (5.6) and (5.7);
    $G_j^{(i+1)}$ based on equation (5.2) and equation (5.8);
  end
end

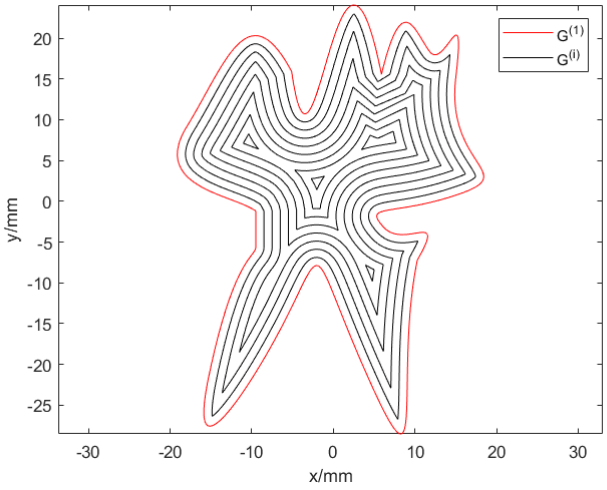


(a) Filling graph with 1mm hatch line spacing (b) Filling partial graph with 0.1mm hatch line spacing

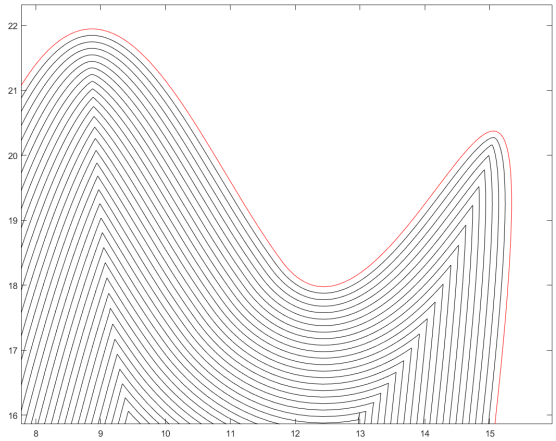
Figure 8: Zigzag parallel hatch of the single-layer contour pattern

Table 2: calculation results

			L/mm	**N**	**t\ms**	**r_{21}**
Zigzag parallel hatch	**Single-layer**	1	797	88	289	6.52
		0.1	9922	907	1886	
	Multi-layer	1	1011	126	266	7.34
		0.1	11079	954	1952	
Contour parallel hatch	**Single-layer**	1	402	11	326	12.55
		0.1	1021	85	4090	
	Multi-layer	1	408	12	440	11.27
		0.1	1099	106	5030	



(a) Filling graph with 1mm hatch line spacing



(b) Filling partial graph with 0.1mm hatch line spacing

Figure 9: Contour parallel hatch of the single-layer contour pattern

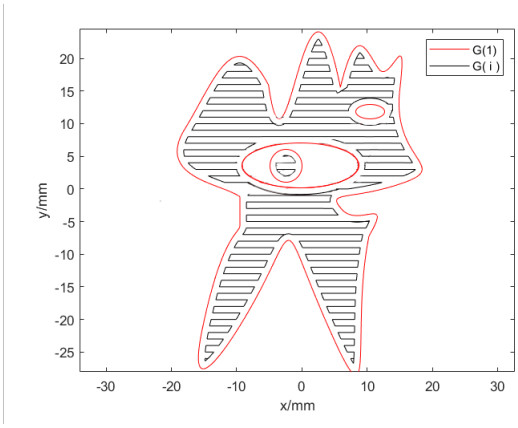


Figure 10: Zigzag parallel hatch of the mutli-layer contour pattern

7 Test the Model

7.1 Analysis of Algorithm Performance

In order to evaluate the performance of the hatching algorithm, we comprehensively consider the running time, efficiency and image distortion of the actual marking process, and construct the following algorithm evaluation function:

$$\psi(t_A, Im, E) \tag{7.1}$$

7.1.1 Definition of t_A

We learned by consulting the literature that in the actual operation of laser marking, the marking time can be expressed as

$$t_A = t_1 + t_2 + t_3 + t_4 + t_5 \tag{7.2}$$

where t_1 represents beam opening extension time,t_2 represents beam off delay, t_3 represents the end delay time,t_4 represents marking time,t_5 represents idle time. We will discuss $t_1 \sim t_5$ separately using the zigzag parallel hatch as an example below.

1.$t_1 \sim t_3$: According to relevant data, the scanning area of the laser marker's working area is within a square area with a side length of about 100mm. The working parameters are roughly as follows: scanning distance 40cm, marking speed 5000mm/s, idle speed 7000mm/s, beam opening extension time t_1 150us, beam off delay t_2 0us, end delay time t_3 50us. So we can consider $t_1 \sim t_3$ as three constants.

$$t_1, t_2, t_3 = Const \tag{7.3}$$

2.t_4: According to the assumptions of the model, we think there will be no equipment failure during actual laser marking. Therefore, the marking time t_4 is actually the sum of the program running time t_p calculated in Chapter 4 or Chapter 5 and the laser movement time. In addition, due to the inflection point delay in the actual marking process, the delay is proportional to the number of horizontal lines.

$$t_4 = t_p + k_1 \times L + k_2 \times N \tag{7.4}$$

$$k_1 = \frac{1}{v_m} \tag{7.5}$$

where v_m is the velocity of laser movement,L is the total length of hatching lines of the hatched curves subject to zigzag parallel or contour parallel hatch

3.t_5:The idle time of the laser refers to the time it takes to transfer from the end of one hatching line to the beginning of another hatching line, obviously it is directly proportional to N and d, and inversely proportional to the idle speed.

$$t_5 = k_3 \times \frac{Nd}{v_i} \tag{7.6}$$

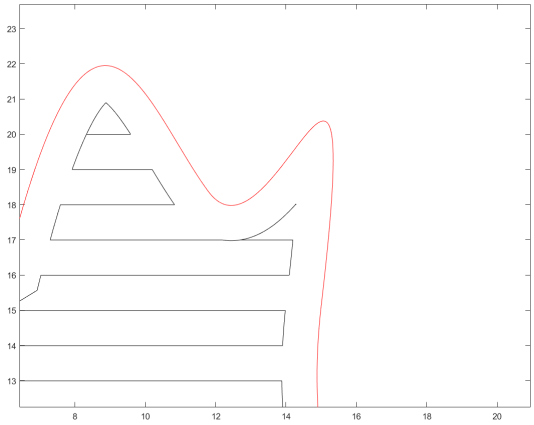
In summary, we get the relationship between t_A and each parameter as follows :

$$t_A = k_1 L + (k_2 + k_3 d)N + t_p + Const \tag{7.7}$$

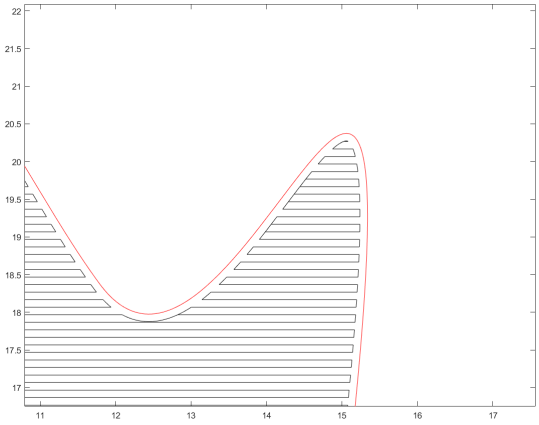
7.1.2 Assessment of Im

It can be seen from the equation 7.7 that when the marking speed is faster, the actual compliance time will be shorter, which will significantly improve the marking efficiency. However, the marking speed cannot be increased indefinitely. When the marking speed exceeds a certain threshold, the energy emitted by the laser per unit time will be significantly reduced, which will affect the clarity of the marking pattern.

What's more, when conditions permit, the larger the hatching line spacing of laser marking, the shorter the total length of hatching lines, and the higher of the efficiency. However, at certain boundaries with larger curvature, if the hatching line spacing is greater than a certain value, the hatching line will appear image distorted as shown in the figure 11



(a) Distortion caused by hatching line spacing of 0.1mm



(b) Normal situation with hatching line spacing of 0.01mm

Figure 11: Image distortion

So we qualitatively give the functional relationship of Im as follows:

$$Im = Im(v_m, d) \tag{7.8}$$

7.1.3 Assessment of E

From the question, we know that the contour lines of any shape have extremely high requirements on the running efficiency of programs. Therefore, on the basis of ensuring the integrity of the image information, use horizontal lines to fill as much as possible.

Besides, as the marking area increases, the laser transmission distance becomes longer, and the beam focusing performance will be greatly reduced. At this time, the power of the laser needs to be increased to ensure the clarity of the image.

Similarly, we give the functional relationship of E as follows:

$$E = E(Lt, v_m, S) \tag{7.9}$$

where v_m is the velocity of laser movement, S is the area enclosed by the contour curve,Lt:

$$Lt = \begin{cases} 1, & \text{choose zigzag parallel hatch} \\ 0, & \text{choose contour parallel hatch} \end{cases} \tag{7.10}$$

In summary, we get the performance function for evaluating the incubation algorithm

$$\psi(t_A, Im, E) = P(t_A(L, N, d, , t_p), Im(v_m, d), E(Lt, v_m, S)) \tag{7.11}$$

7.2 Strategy for Optimizing the Performance and Efficiency of the Hatching Algorithm

In order to better optimize the performance and efficiency of the hatching algorithm, we choose genetic algorithm to find the optimal solution. Genetic algorithm is a computational model that simulates the biological evolution process of Darwinian genetic selection and natural elimination. The specific algorithm steps are as follows:

Step 1: **Determine how chromosomes are coded:**In this paper, the binary coding method is adopted, and two different hatch methods are represented by binary codes to form substrings, and then the substrings are connected into a chromosome string.

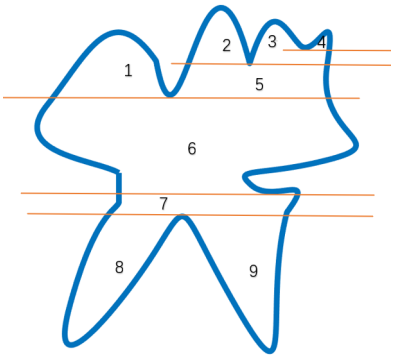


Figure 12: Area division

Step 2: **Initialize the population:**After randomly generating a binary code of gene length, there is no need to check whether the binary code meets the requirements after transcoding in this question, so it can be included in the initial population.

Step 3: **Determine fitness function:**There are three optimization goals t_A, Im, E in this question, which are solved by converting equation 7.11 into single-objective optimization problems.

$$min \quad P(x) \qquad\qquad x \in \Omega \tag{7.12}$$

Step 4: **Screen outstanding individuals:**The selection of excellent individuals is based on the roulette method. In the roulette method, the probability of each individual being selected for inheritance is proportional to the value of its fitness. The higher the fitness, the greater the probability of being selected; the lower the fitness, the smaller the probability of being selected. In this question, we use marking time as a parameter to select this value:

$$sumFitness = \sum t_A \tag{7.13}$$

The selection probability of a certain body i in the population:

$$Ps_i = \frac{t_{Ai}}{sumFitness} \tag{7.14}$$

It can be seen from the formula that the probability Ps reflects the proportion of the fitness of a single individual to the total fitness of all individuals in the entire population. If the fitness of the individual is greater, the probability of him being selected for inheritance is higher, and if the fitness of the individual is smaller, the probability of him being selected for inheritance is lower. The result and purpose of this calculation are: the best chromosomes are copied into multiple points and passed on to the next generation, the middle chromosomes maintain the original level, and the poor chromosomes are not selected and eventually eliminated.

Step 5: **The design of genetic operators:**After determining the binary encoding of chromosomes, consider the crossover and mutation operations of chromosomes.

Step 6: **Set relevant parameters:**The genetic algorithm itself has three parameters, namely the crossover probability P_c, the mutation probability P_m and the maximum genetic algebra P_d. Generally speaking, the crossover probability of 0.6 can ensure the full evolution of the population. the probability of mutation is small, so we set the probability of mutation is 0.005, which is more in line with the laws of nature. The maximum genetic algebra is 200 to ensure that the optimization results fully converge.

We get the optimal individual through the genetic algorithm, in other words, we could use different hatching methods for different areas by dividing the area, this has an important application in actual production: When marking a large number of contour patterns of the same kind, it can be operated in an assembly line, for a specific laser marker, let it only perform the same hatch method for the specific area. Such an assembly line operation method not only improves work efficiency but also ensures the clear and complete filling curve.

8 Strengths and weaknesses

8.1 Strengths

- In the zigzag parallel hatch method, the scan line filling algorithm in the polygon filling algorithm based on computer graphics is improved, making it suitable for convex polygons, concave polygons, and mutually nested multi-layer polygons when there are many contour points on the given boundary filling of field graphics.
- The analogy is based on the polygon filling algorithm of curve integral and Green's formula, which realizes the discovery and elimination of sharp points by judging the detour direction of the path, and finally realizes the split and merge of the filled area.
- In the contour parallel hatch algorithm, the angle discrimination method is used to judge whether the point is inside the polygon, and then to judge when the contour curve reaches the end point

8.2 Weaknesses

- When the genetic algorithm is applied, the population is small and it is difficult to find the optimal solution.

References

[1] N. Raghunath, Pulak M. Pandey. Improving accuracy through shrinkage modelling by using Taguchi method in selective laser sintering. 2006, 47(6):985-995.

[2] Ismail Al-Rawi. Implementation of an Efficient Scan-Line Polygon Fill Algorithm. 2014,

[3] Chongyang Deng, Qingjun Chang, Kai Hormann Iterative coordinates[J] Computer Aided Geometric Design, 2020, 79

[4] Wei Qing Wang. The Filling Algorithm for Scanning Based on the Chain Structure. 2011, 1286:404-409.

[5] Yu Yasufuku Integral points and relative sizes of coordinates of orbits in $\mathbb{P}^N$[J] Mathematische Zeitschrift, 2015, 279(3-4)

[6] Jia Xing Xu, Gang Li, Ji Yao Wang, et al. New Method for Polygonous Node Automatic Creation Based on Triangulation. 2013, 2301:2442-2445.

Appendices

Appendix A Tools and software

Paper written and generated via LaTeX, free distribution.

Graph generated and calculation using MATLAB R2019a.

Appendix B The Codes

Due to space limitations, only part of the code is given here, please refer to the supporting material for the detailed code.

Make circle

```
function y=makecircle(r,n)  %r is the radius;
%n is the number of equal parts of the circle.
    m=2*pi/n;
    y=zeros(n,2);
    for i=1:1:n
        y(i,:)=[r*cos(m*i),-r*sin(m*i)];
    end
    axis equal;
end
```

Inward

```
function y=inward(all,d)

   while (all(1,:)==all(end,:))
       all(end,:)=[];
   end
    n=size(all,1);
    L1=zeros(n,2);
    a=(all(2,:)-all(n,:));
    b=zeros(2,1);
    b(2)=-a(1);
    b(1)=a(2);
    b=b./(sum(b.^2)^(1/2));
    b=b';
    L1(1,:)=all(1,:)+b*d;
    note=[];
    note1=[];
    for i=2:1:n-1
        a=(all(i+1,:)-all(i-1,:));
        b=zeros(2,1);
        b(2)=-a(1);
        b(1)=a(2);
        b=b./(sum(b.^2)^(1/2));
        b=b';
        L1(i,:)=all(i,:)+b*d;
        if(i>=4)
            note=[note;findwr(L1,i)];
        end
        if (comp2(all(i-1,:)',all(i,:)',L1(i-1,:)',L1(i,:)'))
            note1=[note1,i];
        end
    end
    a=(all(1,:)-all(n-1,:));
    b=zeros(2,1);
    b(2)=-a(1);
    b(1)=a(2);
    b=b./(sum(b.^2)^(1/2));
    b=b';
    L1(n,:)=all(n,:)+b*d;
    note=[note;findwr([L1;L1(1,:)],n+1)];
    L1=[L1;L1(1,:)];
```

```
    if (note)
        L12=resetp(L1,note,note1,all);
    else
        L12={L1};
    end
    for i=1:1:length(L12)
        L112=L12{i};
        if size(L112,1)>2
            L112=[L112;L112(1,:)];
            plot(L112(:,1),L112(:,2),'-k');
        end
    end
    axis equal;
    y=L12;
end
```

Reset Point

```
function y=resetp(L1,note,note1,yall)
    n0=size(L1,1);
    n1=size(note,1)
    mark=ones(n1,1);
    m=1;
    k=note(n1,1);
    if (n1>=2)
        com=note(1:end-1,1)-note(2:end,1);
        for i=n1-1:-1:1
            if (com(i)>0)
                m=m+1;
                mark(i)=m;
                k=[k,note(i,1)];
            else
                while(m>0  && note(i,1)<k(end))
                    k(end)=[];
                    m=m-1;
                end
                m=m+1;
                mark(i)=m;
                k=[k,note(i,1)];
            end
        end
    else
        if(n1==1)
            mark=1;
        end
    end

        deco=judgcc(L1,mark,note);
        ano=deco{3};
        dele=[deco{1},note1];
        comple=deco{2};
        L1(comple(:,3),:)=comple(:,1:2);
        i=1;
        remjd=[];
        while(i<=n1)
            j=judgct(note(i,:),note1,n0);
            note1=j{2};
            if(j{1})
```

```
            remjd=[remjd,note(i,1)+2:note(i,2)-1];
        end
        i=i+1;
    end
    m=size(ano,1);
    numc=(1:size(L1,1))';
    Lc=[L1,numc];
    Lc([dele,remjd],:)=[];
    numc2=(1:size(Lc,1))';
    y={};
    kk=Lc(:,3);
    tt=Lc(:,1:2);
    for i=1:1:m
        a=numc2(kk==ano(i,1));
        b=numc2(kk==ano(i,2));
        c=numc2(kk==ano(i,3));
        d=numc2(kk==ano(i,4));
        y=[y,{tt([a:b,c:d],:)}];
        Lc([a:b,c:d],:)=[];
    end
    tt=Lc(:,1:2);
    y=[y,{tt}];
    for i=1:1:length(y)
        m=length(y{i});
        if(m<=3)
            y{i}=[];
        elseif(m<=200)
            yy=y{i};
            j=1;
            while (j<=m)
                if(deld2(yall,yy(j,:)))
                    j=m+1;
                    y{i}=[];
                end
                j=j+1;
            end
        end
    end
end
```