International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n2.26
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Research and Application of Euler Cutting Path Based Optimisation Algorithm with Greedy-RRT Algorithm

Yongheng Shen *

School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo, 255000, China

*Corresponding Author: 18678580683@163.com

ABSTRACT

With the development of social economy, mould processing enterprises have an increasingly strong demand for reducing costs and improving economic efficiency. The optimisation of steel plate cutting path is one of the key links to improve production efficiency. In this paper, the optimal cutting path problem of steel plate under different cutting tasks is investigated, and the optimal cutting path under complex geometries is found through various optimisation algorithms. For cutting task N3, the cutting path is divided into outer contour cutting, inner hole cutting and inner hole part cutting. According to the problem, all rectangular parts inside the ellipse need to be cut before the ellipse. Through the Eulerian cutting path optimisation algorithm, we solve the air-travel problem between the inner holes and the outer contour as well as the air-travel problem between the inner holes of the ellipse and the parts inside the ellipse. The total length of the finally determined optimal air-travel path is 97 units. By combining the greedy algorithm and the RRT algorithm, we determine the number and location of the bridges and design the optimal cutting path starting from the lower right corner of the steel plate. The final clearance of the bridges is 4 units, the total length of the combined optimal path clearance is 35.56 units, and the number of bridges is 8, which are located between adjacent rectangles.

KEYWORDS

Plate cutting path optimisation; Shortest path for air travel; Eulerian cutting paths; Greedy algorithm; RRT algorithm; Bridge Design

1. INTRODUCTION

With the rapid development of social economy, mould processing enterprises are facing the double pressure of cost reduction and economic efficiency improvement in the fierce market competition. In this context, improving production efficiency has become one of the key factors for the sustainable development of enterprises [1]. Steel plate cutting as an important link in mould processing, its efficiency directly affects the cost and time of the whole production process. Therefore, optimising the steel plate cutting path and reducing the length of the empty journey has become an important means to improve production efficiency [2]. Steel plate cutting path planning is a complex optimisation problem, involving various factors, such as cutting sequence, air travel distance, equipment utilisation, etc. In actual production, due to the discontinuity of the boundary and the bore, the cutting process will inevitably produce empty travel. These empty journeys not only increase the cutting time, but also may lead to material waste and increased energy consumption. Therefore, finding the optimal cutting path to minimise the length of the air-travel is of great significance to improve the production efficiency and reduce the production cost. In recent years, many researchers have proposed various methods to solve the steel plate cutting path optimisation problem. Although the traditional enumeration method can find all possible paths and select the optimal path from them, it has high computational complexity and is suitable for small-scale problems. For more complex cutting tasks, nonlinear programming models and other advanced optimisation algorithms are widely used to solve more efficient cutting paths..

2. MODEL ASSUMPTIONS

The data used in this paper comes from Question A of the May 1 Mathematical Modelling Competition 2024. To construct a more precise mathematical model, we make the following reasonable assumptions or constraints based on the actual situation:(1) During the steel plate cutting process, we ignore any errors caused by the machine's lack of precision; (2) We ignore the errors caused by the volume of the cutting tool and abstract the tool as an ideal point; (3) During the cutting process, the steel plate does not move, deform, or undergo any other changes.

3. THE SHORTEST DISTANCE BETWEEN THE OUTER CONTOUR AND THE RECTANGULAR PART SOLUTION

Priority cutting rectangular parts and then oval hole cutting, and follow the principle of the shortest air distance, through the Euler cutting path [3] optimisation algorithm to find out the closest distance to cut rectangular parts, combined with the problem solved in problem two of the outer contour to the four circular holes in the air distance and the outer contour and the rectangular parts of the shortest air distance between the parts, you can find the optimal path. According to the dimensional annotation shown in the figure, the shortest air distance is the distance between the lower left vertex of rectangle 01 in the lower left corner vertically downwards and the intersection point of the outer contour. The distance is s = (80/2) - 5 - (3/2) = 18.5.

4. SOLVING FOR THE SHORTEST AIR DISTANCE BETWEEN RECTANGULAR PARTS

The distribution of the 12 rectangular parts is shown as follows:

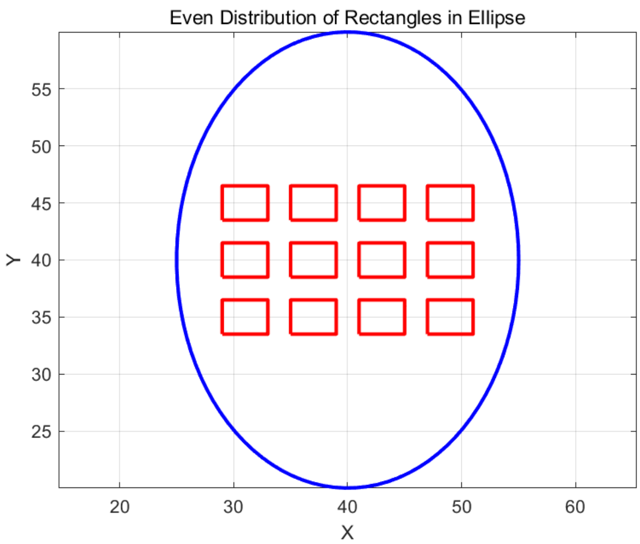

Figure 1. Distribution of 12 rectangular parts

Each steel plate can be considered as a vertex in the graph and the cutting scheme corresponds to an edge in the graph. The goal is to find a path such that each edge belongs to at least one path and the length of each path satisfies specific requirements.

4.1. Definition of Unknowns

(1) Define the four vertices of a rectangular part as nodes, denoted as V.

(2) Define the edge between two nodes as the path connecting these two rectangular parts, denoted E.

(3) Define the degree of each node as d, the number of edges connected to that node.

(4) Define the degree at node V_i to be d_i.

(5) Define the length of the Eulerian path as L, i.e., the number of edges passing through.

(6) $X_{i,j}$: indicates whether the rectangular parts in row i and column j are cut or not, and takes the value of 0 or 1.

(7) $y_{i,j}$: indicates the number of rows from row i, column j to the next cut rectangular parts, taking the value of 1 (above), 2 (right), 3 (below), 4 (left)

4.2. Constraints

(1) The four edges of each rectangular part must be cut once.

(2) Cutting starts from the lower right corner of a rectangular part closed at the lower right corner and ends back at that point.

(3) The cutting path must be continuous, i.e., the rectangular parts cannot be skipped.

4.3. Solving Process

(1) Construct a graph-theoretic model: abstract the problem into a graph-theoretic problem, where each rectangular part vertex is a node, and the distance between two rectangular parts is the edge connecting these two nodes. Construct a graph G=(V, E), where V is the set of rectangular promenades and E is the set of edges connecting these rectangular parts.

(2) Determine whether Euler's path exists: according to the properties of Euler's path, the condition for the existence of Euler's path is:

a. The graph is connected: that is, a path exists between any two nodes.

b. The degree of all nodes is even, or there are and only two nodes with odd degree, and these two nodes that is the beginning and end of the path.

(3) Finding Euler's path: Use Fleury's algorithm to ensure that you pass through each edge once and only once to satisfy the requirements of the problem.

(4) Path optimisation: start from the lower right point of the rectangle and return to that point. After finding the Eulerian path, the path is optimised to ensure that the start and end points are correct and that the path has the shortest length.

(5) Output optimal path: output the optimised optimal path, i.e. the best cutting route for cutting rectangular parts.

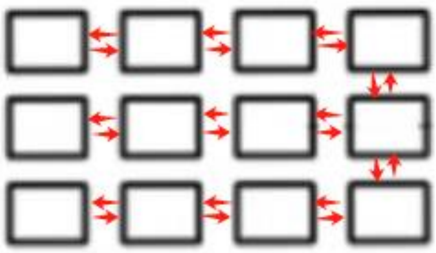

Figure 2. Cutting rectangular optimal cutting path

The shortest clearance for the rectangular part has been marked with a red arrow in the above figure, and the resulting shortest clearance is 44.

5. OUTER CONTOURS, RECTANGULAR PARTS AND ELLIPTICAL HOLES TO SOLVE THE SHORTEST AIR DISTANCE

According to the above solved cutting path, cutting rectangular parts return tool should first cut the elliptical hole, and then return to the outer contour to continue cutting, to cut the starting point as the origin.

L_i represents length of air travel; (x_{jk}, y_{jk}) represents coordinates of any point on the outer contour (i=1,2,... .n); (x_{mn}, y_{mn}) represents coordinates of points on the ellipse $\frac{(x+40)^2}{225}+\frac{(y-40)^2}{400}$=1.

The air distance from point a to any point on the ellipse is S1 and the air distance from that point on the ellipse to the outer contour is S2.

$$L_i = S1 + S2$$

The computational solution is performed using matla code. L_i is minimised to 18.5 when the line between a and a point on the elliptical hole and a point on the outer contour is a vertical line. The air travel (double travel) between the outer contour to the circular bore is 16. So the total length of the air travel is 18.5+44+18.5+16=97. The optimum route is shown in the following Figure 3:

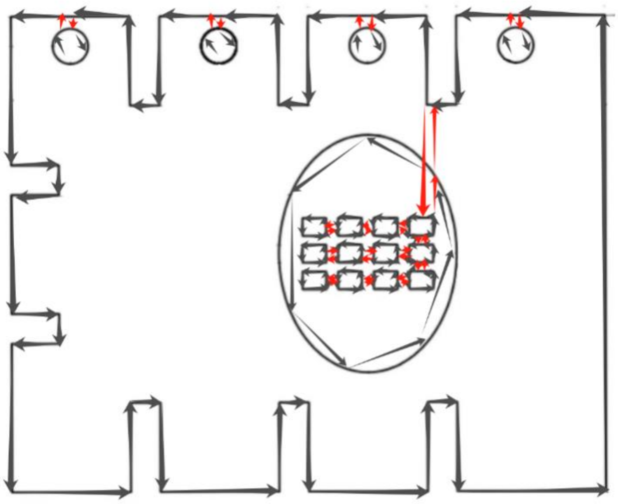

Figure 3. Optimal route for cutting task 3

6. FINDING THE SHORTEST BRIDGE CLEARANCE DISTANCE BETWEEN RECTANGULAR BLOCKS

Following the principle of the shortest air distance, but also to ensure the integrity of the four rectangular parts, so to carry out the 'bridge' design, and then determine the number and location of the bridge between the four small rectangular parts. Considering the bridge width and air distance, the greedy algorithm [4] and RRT algorithm combination, in order to achieve to meet the machining requirements under the premise of finding the optimal machining path.

6.1. Definition of Unknowns

Define the position coordinates of any point on the edge of the rectangle as (i=1, 2, 3....). The air distance between rectangles is s, cutting the total air distance is s, the distance between two points on the rectangular edge is L.

6.2. Determine the Constraints

(1) Perform bridging to ensure the integrity of the rectangular parts;

(2) Cutting paths should be continuous to ensure that adjacent rectangular parts for bridging design.

6.3. Model Building

The RRT (Rapidly-exploring Random Trees) algorithm is an effective path planning and searching algorithm widely used in robot navigation and other engineering fields such as animation, spatial design and automatic control systems. It is particularly suitable for dealing with path planning problems in high-dimensional spaces and complex obstacle environments. The algorithm is suitable for path optimisation in high-dimensional complex environments due to its strong search capability, few parameters and fast path search. In the process of expanding the spanning tree, the expanded nodes are randomly sampled, so its search path can only find the path between two points, and can not get the optimal path and redundant nodes of the large amount of computation, planning time, in order to solve the traditional RRT algorithm brought about by the non-optimal path planning, slow search speed, this introduces a greedy algorithm in the Kruskal algorithm to select the optimal solution of each subproblem and discard the other solutions in order to solve the path optimisation problem in high-dimensional complex environments. The Kruskal algorithm in the greedy algorithm is introduced in this regard to select the optimal solution of each sub-problem and discard the other solutions to reduce the computation of redundant points [4]. The optimisation of the RRT algorithm path planning is mainly carried out in the following five steps:

(1) Random tree initialisation

When the algorithm is reading the coordinate system, the current position is used as the search starting point of the fast expansion tree, and the received destination coordinate point is used as the end point of the fast expansion tree. The initialisation of the random tree includes the root node and the end point with a threshold.

(2) Perform random sampling

The expansion of the random tree uses random sampling to detect whether there is a collision at a node and thus determine whether there is an obstacle at that point [5]. The distance between any two points of the rectangular edge is calculated using Euclidean distance, which is shown in Eq. in 2D environment:

$$L = \sqrt{(X_1 - X_2)^2} + \sqrt{(Y_1 - Y_2)^2}$$

After sorting the calculated distances, the node with the shortest distance is selected for collision detection (to determine whether there is an obstacle between the 2 nodes). If there is no obstacle to collide, then the node is added to the fast expansion tree; if the node has a collision event, the algorithm determines that there is an obstacle at the point, then regenerate the random point, and then recycle the detection of the Euclidean distance calculation and judgement.

(3) Expansion of nodes

The expansion of nodes shall be based on the previous step to detect collision events, and nodes without collision events are added to the random tree to expand the random tree.

(4) End point stop expansion

An end point with a threshold value is required for the initialisation of the random tree. When the Euclidean distance between a node of the expanded tree and the end point is less than the set threshold, the program judges that the node is the end point and stops the expansion of the random tree. After completing the above steps 1 path from the starting point to the end point can be obtained. Accordingly the optimal roadmap can be obtained as shown in Figure 4.

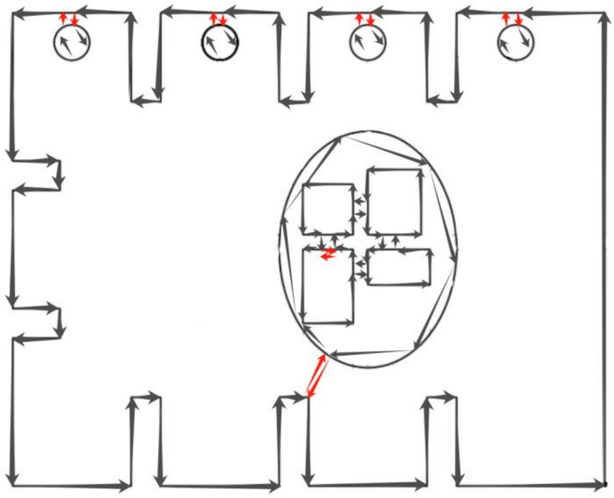

Figure 4. Optimal route for cutting task 4

In this regard, we can find out the minimum distance between the rectangles s that is the width of the two sections of the bridge 4, and then combined with the results of the problem 2 to sum up the final result S is 31.56 + 4 = 35.56. In addition, we can see that the number of bridges from the figure is 8, located in the adjacent rectangles between the two.

(5) Handling redundant points

The redundant points will be processed by removing them.

7. CONCLUSION

In this paper, through the study of steel plate cutting path optimisation, the method of finding the optimal path under different cutting tasks is proposed, and significant optimisation results are achieved. In the cutting task N3, the cutting path is divided into outer contour cutting, inner hole cutting and inner hole parts cutting. According to the question, all rectangular parts inside the ellipse need to be cut before the ellipse. Through the Eulerian cutting path optimisation algorithm, we solved the problem of the air distance between the inner holes and the outer contour as well as the air distance between the inner holes of the ellipse and the parts inside the ellipse. Specifically, the total length of the air-travel between the rectangular parts is 44 units, the air-travel (double-travel) between the outer contour to the circular inner hole is 16 units, the air-travel (double-travel) between the 12 rectangular parts and the elliptical inner holes and the outer contour is 37 units, and the combined optimal air-travel path has a total length of 97 units. By combining the greedy algorithm and the RRT algorithm, we determine the number and location of the bridges and design the optimal cutting path starting from the lower right corner of the steel plate. The resulting bridge clearance is 4 units, the total combined optimal path clearance is 35.56 units, and the number of bridges is 8, which are located between adjacent rectangles.Through these studies, we not only provide specific optimisation methods, but also provide theoretical support and technical guidance for the path planning of mould processing enterprises in actual production. These methods help enterprises to achieve cost reduction and benefit enhancement under the premise of ensuring cutting quality. In addition, the research results of this paper also provide reference and reference for further research in related fields. By optimising the steel plate cutting path, enterprises can significantly improve productivity and reduce resource waste, thus maintaining competitive advantages in the fierce market competition.

REFERENCES

[1] Amine K, Djellab R. Industrial and Urban Applications of Eulerian and Chinese Walks[M]//Graph Theory for Operations Research and Management: Applications in Industrial Engineering. IGI Global, 2013: 271-279.

[2] Makarovskikh T A, Panyukov A V, Savitskiy E A. Mathematical models and routing algorithms for economical cutting tool paths [J]. International Journal of Production Research, 2018, 56(3): 1171-1188.

[3] Makarovskikh T A, Panyukov A V. Mathematical model for a cutting path avoiding intersections [J]. IFAC-PapersOnLine, 2020, 53(2): 10455-10460.

[4] Eapen N A, Heckendorn R B. Cutting path optimization for an automatic cutter in polynomial time using a 3/2 approximation algorithm [J]. The International Journal of Advanced Manufacturing Technology, 2021, 113: 3667-3679.

[5] Makarovskikh T A. On the number of starting points for a fixed cutting plan and fixed cutter trajectory[C]//Intelligent Technologies for Information Processing and Management (ITIPM'2014). 2014: 239-244.