Route Planning through Genetic Algorithm for Multi-Axis Motion Control

Ting-Ching Lee
Department of Power Mechanical Engineering
National Tsing Hua University
Hsinchu 30013, Taiwan
s111033626@m111.nthu.edu.tw

Jia-He Tee
Department of Computer Science
National Tsing Hua University
Hsinchu 30013, Taiwan
s112062806@m112.nthu.edu.tw

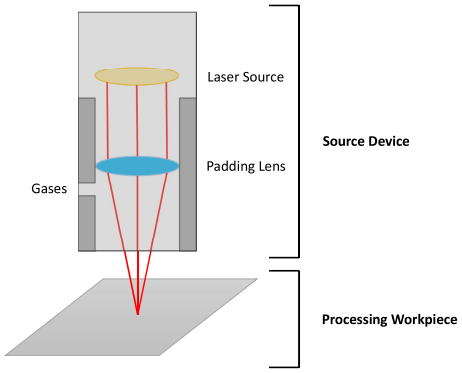
Chuan-Kang Ting
Department of Computer Science
National Tsing Hua University
Hsinchu 30013, Taiwan
ckting@cs.nthu.edu.tw

***Abstract*—Route planning is crucial in the manufacturing industry, especially in cutting systems. Multi-axis motion control is pivotal to the laser cutting process in that it substantially enhances the efficiency by moving the source device along two axes simultaneously to target the workpiece. However, optimizing the routes for multi-axis motion control remains a complex challenge due to the need for coordination and efficiency. This study formulates it as a close-enough traveling salesman problem (CETSP) and designs a genetic algorithm (GA) for solving this routing problem. First, the line simplification algorithm is employed to convert the cutting pattern into a series of points, serving as the city points in the CETSP. Second, a structural approach is presented to address the difficulty in determining the turning points. For the GA, we developed crossover and mutation operators to improve its effectiveness and efficiency. The experimental results show that the proposed GA reduces route lengths by 17–28% and improves the variable speed in the route by 47–69% on two test cutting patterns.**

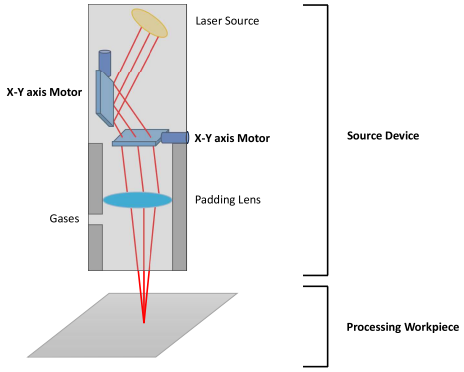
***Index Terms*—Route planning, cutting process, evolutionary computation, genetic algorithm, multi-axis motion control, line simplification algorithm, close-enough traveling salesman problem.**

I. Introduction

With market expansion and a surge in demand for personalized processing of workpieces, the manufacturing industry faces a significant challenge in route optimization. This problem stems from two distinct manufacturing equipment systems, i.e., marking process [1] and multi-axis motion control [2] (Fig. 1). These two systems require different routing strategies for processing workpieces. The marking processing system involves manufacturing by tracing the pattern within a figure on a workpiece. In this system, a route corresponds to the pattern, and its optimization will focus on determining the optimal pierce points for the starting and ending points, the tool trajectory, and the airtime motion. On the other hand, the multi-axis motion control system ordinarily comprises a source motor and a x-y axis motor. For instance, in the laser cutting process, the former moves the source device, and the latter emits laser beams to specific workpiece positions. These controllers operate simultaneously but at different speeds: the source motor usually move slower due to the movement of entire source device, whereas the x-y axis motor operates at a higher speed as it only adjusts the angle of the padding glass.



(a) Marking process [3]



(b) Multi-axis motion control [4]

Figure 1: Two types of manufacturing systems

This study presents a novel way to address the route optimization problem in multi-axis motion control systems. First, the routing problem is formulated as a close-enough traveling salesman problem (CETSP). Second, we design a genetic algorithm (GA) to solve this problem. A major difference between a marking process system and a multi-axis motion control system lies in the presence of an x-y axis motor: the latter includes the x-y axis motor, while the former does not, as demonstrated in Fig. 1. Additionally, in the multi-axis motion control system, the use of padding glass in the x-y axis motor enables that the source motor does not need to pass through

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00221

the designated pattern on the workpiece; instead, the solution incorporates turning points that originate from the initial vertex defined by a radius. The line simplification algorithm (LSA) is employed to select the points as the cities for the CETSP from the pattern and to determine the permutation of cities. The route optimization then focuses on finding the permutation of turning points that minimizes the total route distance. Notably, the movement speed can vary along the route in the multi-axis motion control system. This study considers the speed information of source device in the chromosome representation and develops crossover and mutation operators to improve the GA. A series of experiments is conducted to examine the effectiveness and efficiency of the proposed method.

The remainder of this paper is organized as follows: Section II reviews the related studies. Section III elaborates the proposed method. The experimental results are presented in Section IV. Finally, Section V gives the concluding remark.

II. Related Work

In the manufacturing industry, there are two primary processing systems based on the movement of the source device along its axis: the marking processing system and the multi-axis motion control system. In the marking processing system, the optical lens remains stationary inside the source device, and the source device is responsible for processing the workpiece by moving itself along its axis. The goal of this system is to adjust the speed fluently. On the other hand, the multi-axis motion control system will move both x-y axis motors to process the workpiece. Unlike the marking processing system, the route in the multi-axis system is not constrained to following the pattern alone, since the padding glass can be set at various angles to accomplish the desired pattern. The goal of this system is to optimize both the route and speed in each path [5].

As in the marking processing system, computer numerical control (CNC) cutting addresses the shaping of metal or plastic workpieces using various tools such as lasers, plasma, gas, or water-jet [5, 6]. CNC cutting involves considerations such as pierce points, lead-in and lead-out points that facilitate the tool trajectory from switching the tool off, and airtime motions [7, 8]. Nonetheless, the multi-axis motion control system holds the advantage of not requiring consideration for tool radii. This advantage gives rise to the following scenarios:

- The pierce point of a pattern serves as both the starting point for the routing problem and the lead-in point for the tool trajectory.
- The point of switching the tool off is the ending point of the route problem, and it is also the lead-out point of the tool trajectory.
- The tool trajectory aligns with the pattern itself.
- The airtime motion entails direct movement since there are no intersecting routes between the tool and the workpiece.

The cutting route optimization problem needs to derive pierce points and tool trajectory from the given pattern and then search for a route of the source device that satisfies the objective (e.g., shortest length) and constraints. Current studies on cutting route optimization mostly focus on the marking problems, while route optimization for multi-axis motion control systems are under-addressed. In cutting route optimization, the pattern is typically decomposed into a series of points, analogous to cities in the traveling salesman problem (TSP). These points closely follow the contour of the pattern, establishing an initial route for the routing problem. In the multi-axis motion control system, there is a padding lens inside the source device that will operate within a certain range of the padding radius on the workpiece. The route of the source device is not required to follow the pattern.

The close-enough traveling salesman problem (CETSP) is a variant of TSP that associates of each node with a radius [9]. Its objective is to find the shortest tour starting from a depot, traversing each radius around the nodes (cities) and returning to the depot. In the CETSP, routes are not required to visit city points directly; instead, the problem only necessitates that routes visit turning points located within the radius of city points. Various algorithms, including exact algorithms [10, 11], branch-and-bound algorithms [12], and heuristic algorithms [13, 14, 15] have been presented for solving the CETSP [16, 17]:

- Exact algorithms can provide optimal solutions but are often extremely time-consuming due to the intricate calculations involved.
- Branch-and-bound algorithms exhibit a capability to provide near-optimal solutions.
- Heuristic algorithms offer near-optimal solutions with shorter computation time, making them suitable for striking a balance between solution quality and computation time.

III. Methodology

This study aims to solve the route optimization problem in multi-axis motion control systems. To this end, we formulate the routing problem as a CETSP and develop a GA for addressing this problem. The following describes the proposed methods in detail.

A. Problem Formulation

The cutting pattern can be represented as a directed graph $G_P = (V_P, E_P)$, where the vertex set $V_P = \{v_1, v_2, \ldots, v_n\}$ includes the points decomposed from the pattern, and the edge set $E_P = \{(v_i, v_{i+1}) | v_i, v_{i+1} \in V_P\}$ contains edges (v_i, v_{i+1}) from vertex i to $i+1$. Similarly, a route of the source device can be represented by a directed graph $G_s = (V_s, E_s)$, where V_s is the set of selected turning points, and E_s is the set of paths of the source device. Each edge $e \in E_s$ is associated with a cost $c(e)$, which is determined by the Euclidean distance in this study. This optimization problem is to find a shortest

route G_s that covers all points along the pattern G_p, which can be written as

$$\begin{aligned} \underset{G_s}{\operatorname{argmin}} \sum_{e \in E_s} c(e) \\ \text{s.t. } \operatorname{Covg}(G_s, G_P) = 1 \\ \operatorname{InR}(G_s, G_P) = 1 \end{aligned}$$

The first constraint indicates that G_s must cover G_P with respect to a working range radius r and distance function $\operatorname{dist}(\,)$. Let V_{NC} be the set of vertexes in the pattern not covered by the source device's route, i.e., $V_{NC} = \{\mu \in V_P | \min \operatorname{dist}(\mu, V_s) > r\}$. The coverage rate can then be computed by

$$\operatorname{Covg}(G_s, G_P) = 1 - \frac{|V_{NC}|}{|V_P|}.$$

Therefore, a route of source device that fully covers the pattern gives $\operatorname{Covg}(G_s, G_P) = 1$.

The second constraint requires that all points to be within the working range. Let $v_i \in V_P$ be the first point going out of the bound defined by its corresponding point in V_s and radius r. If no v_i exists, i.e., every point within bound, then i is set to the cardinality of V_P. The within-radius value is calculated by

$$\operatorname{InR}(G_s, G_P) = \frac{i}{|V_P|}.$$

Accordingly, a route with no point out of bound has a within-radius value $\operatorname{InR}(G_s, G_P) = 1$.

B. Line Simplification Algorithm

This study employs the line simplification algorithm (LSA) to reduce the number of points needed for representing a route [18]. Specifically, the Ramer-Douglas-Peucker (RDP) algorithm [19] and Visvalingam-Whyatt (VW) algorithm [20] use distance and area thresholds for reduction.

1) Ramer-Douglas-Peucker Algorithm: The RDP algorithm (see Algorithm 1) is an iterative approach to approximating the polyline. It recursively divides the polyline and retains points that are farther than the threshold to the overall shape of the curve.

2) Visvalingam-Whyatt Algorithm: The VW algorithm (see Algorithm 2) uses successive eliminations to simplify the polyline by calculating the areas formed by three points and then removing the middle point of the smallest area until the threshold area is reached, as lines 12 and 13 in Algorithm 2 indicated.

As above stated, this study adopts both RDP and VW algorithms to allocate the city points for tackling the CETSP.

C. Genetic Algorithm

Regarding the route planning for multi-axis motion control, the presence of padding radius renders a flexibility in planning the route for the source device to trace the pattern. This characteristic is analogous to the relaxation from exactly passing through the city points in the CETSP. Therefore, we formulate this route planning problem as a CETSP.

Algorithm 1 Ramer-Douglas-Peucker algorithm (RDP)

Input: S: set of n points for the figure
ϵ: distance threshold
Output: S_{output}: set of simplified points

1: $d_{max} = 0$
2: $index = 0$
3: $point_{last} = n$
4: **for** $index = 2$ to $(point_{last} - 1)$ **do**
$d =$ perp_dist($S[index]$, sline($S[1]$, $S[point_{last}]$))
5: **if** $d > d_{max}$ **then**
6: $index = i$
7: $d_{max} = d$
8: **end if**
9: **if** $d_{max} > \epsilon$ **then**
10: $S_1 =$ RDP($S[1 : index], \epsilon$)
11: $S_2 =$ RDP($S[index : point_{last}], \epsilon$)
12: $S_{output} = (S_1[1 : point_{last} - 1], S_2[1 : point_{last}])$
13: **else**
14: $S_{output} = (S[1], S[point_{last}])$
15: **end if**
16: **end for**

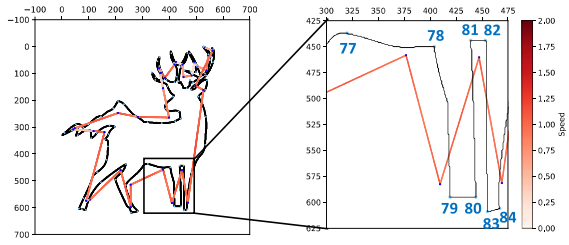
Algorithm 2 Visvalingam-Whyatt algorithm (VW)

Input: S: set of n points for the figure
ϵ: area threshold
Output: S_{output}: set of simplified points

1: **while** true **do**
2: $S_{output} = S$
3: $i = 0$ ▷ index
4: $area = []$
5: $point_{last} = n$
6: **for** $i = 2$ to $(point_{last} - 1)$ **do**
$area[i] =$ triangle_area($S[i-1]$, $S[i]$, $S[i+1]$)
7: **end for**
8: $area_{\min} = \min(area)$
9: **if** $area_{\min} \geq \epsilon$ **then**
10: break
11: **end if**
12: $i =$ index_of($area_{minimum}$, $area$)
13: S_{output}.remove(i)
14: **end while**

In the proposed method, the sequence of visiting city points along with the cutting pattern is determined by the LSA. On the other hand, the permutation of turning points and the movement speed between them are adjustable and require optimization because these factors are crucial for reducing the total route length and smoothing the cutting process. This study designs a GA to optimize the permutation and movement speed due to its recognized performance in combinatorial optimization.

1) Representation: For the GA, a chromosome consists of three pieces of information: positions of LSA points, positions of turning points, and movement speed. First, the LSA points correspond to the cities in the CETSP, which are determined in



LSA points	77	78	79	80	81	82	83	84
LSA position	(320, 437)	(404, 450)	(419, 595)	(444, 595)	(439, 444)	(453, 444)	(455, 609)	(467, 606)
Turning point's position	X	(376, 458)	(409, 582)	X	X	(447, 460)	X	(468, 581)
Speed	X	1	1	X	X	1	X	1.04

Figure 2: Example representation for the reindeer test case.

Algorithm 3 Nearest Algorithm

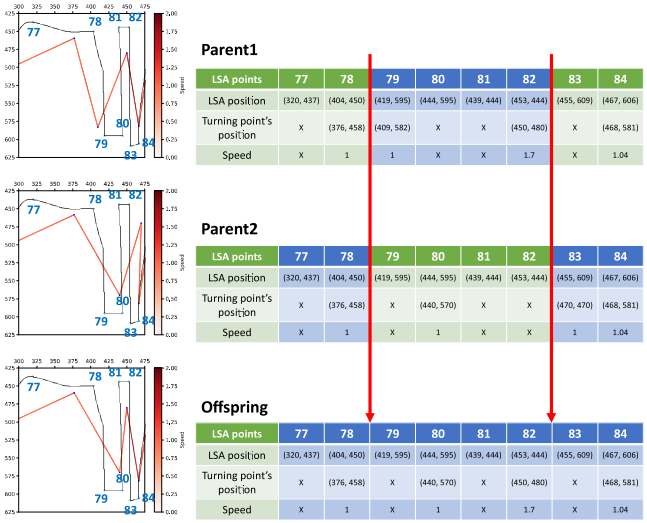
Input: $S_{original}$: set of n points for the figure.
Output: S_{output}: set of n points in sequence for the figure.

```
S_output=S_original[0]
Remove S_original[0] from S_original.
while true do
    Find point_nearest with minimum distance in S_original.
    Appends point_nearest to S_output.
end while
```

the initialization stage. Second, each turning point is within the padding radius of its corresponding LSA point, indicating the location of the route. Third, the movement speed defines the speed for moving from the current turning point to the next. Figure 2 gives an example representation, where the color bar indicates the movement speed, blue points mark the positions of LSA points, and dark blue points mark the positions of turning points.

2) Initialization: In this study, we utilize LSA to sample points as the cities for the CETSP from the pattern points V_p. First, the pattern is converted into a sequence of points using the nearest algorithm (see Algorithm 3). Second, we employ RDP and VW algorithms to derive the simplified points. In these two algorithms (Algorithms 1 and 2), it is essential to segment the pattern and set a threshold for distance or area. This study segments the pattern by 100 points, sets the distance threshold ϵ to 10 mm, and sets the area threshold ϵ to 10 mm^2. The results are denoted by V_{RDP} and V_{VW}, and the union $V_{Union} = V_{RDP} \cup V_{VW}$. Third, a half-padding radius deduction is applied to eliminate the points within a distance shorter than half of the padding radius from their adjacent points. It is worth noting that this process considers the pattern, ensuring the retention of essential points. The final results are called LSA points V_{LSA}, analogous to the cities in the CETSP. Subsequently, the operator generates a population of the specified size and removes two random turning points from the set to increase the diversity.

3) Crossover and Mutation: In this study, we developed crossover and mutation operators to improve the effectiveness and efficiency of GA on the CETSP. The proposed crossover,



LSA points	77	78	79	80	81	82	83	84
LSA position	(320, 437)	(404, 450)	(419, 595)	(444, 595)	(439, 444)	(453, 444)	(455, 609)	(467, 606)
Turning point's position	X	(376, 458)	X	(440, 570)	X	X	(470, 470)	(468, 581)
Speed	X	1	X	1	X	X	1	1.04

Figure 3: Crossover operation on routes.

LSA points	77	78	79	80	81	82	83	84
LSA position	(320, 437)	(404, 450)	(419, 595)	(444, 595)	(439, 444)	(453, 444)	(455, 609)	(467, 606)
Turning point's position	X	(376, 458)	(409, 582)	X	X	X	X	(468, 581)
Speed	X	1	1	X	X	X	X	1.04

LSA points	77	78	79	80	81	82	83	84
LSA position	(320, 437)	(404, 450)	(419, 595)	(444, 595)	(439, 444)	(453, 444)	(455, 609)	(467, 606)
Turning point's position	X	(376, 458)	(409, 582)	X	X	(450, 450)	X	(468, 581)
Speed	X	1	1	X	X	1	X	1.04

LSA points	77	78	79	80	81	82	83	84
LSA position	(320, 437)	(404, 450)	(419, 595)	(444, 595)	(439, 444)	(453, 444)	(455, 609)	(467, 606)
Turning point's position	X	(376, 458)	(409, 582)	X	X	(447, 460)	X	(468, 581)
Speed	X	1	1	X	X	1.7	X	1.04

Figure 4: Three mutation operations on routes.

named improved 2-point crossover, cuts and shifts the segment between two randomly selected LSA points, coupled with the corresponding turning point's position and movement speed. Notably, the preservation of identical LSA points in both parents is essential to prevent infeasible solutions.

Moreover, the proposed combination mutation chooses a gene to perform one of the following three operations at uniform random (cf. Fig. 4):

- Clear: Removes the turning point and speed of one LSA point, separating into two paths. Subsequently, the paths are reconnected based on the permutation of the turning point; the movement speed is recalculated using the information from the previous gene.
- Change position: Shifts the position of one turning point to a random point within the half-padding radius of the

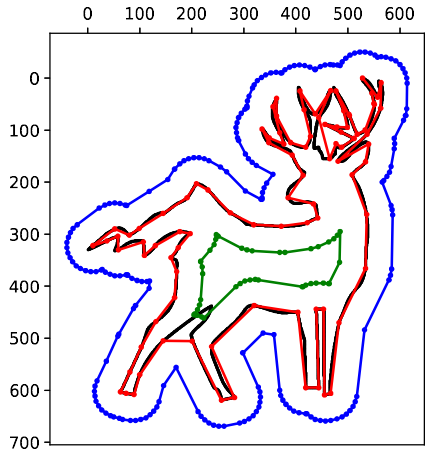


Figure 5: Illustration of coverage rate.

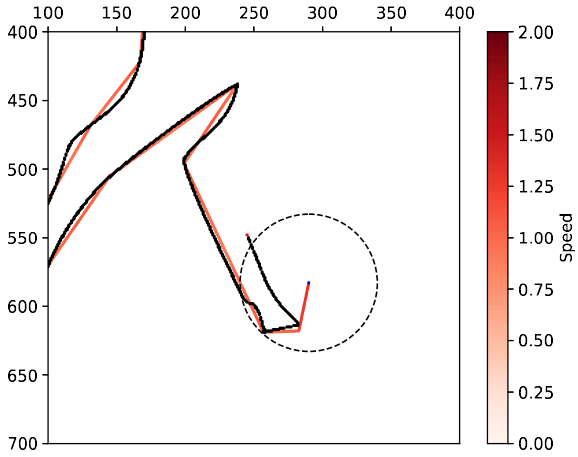


Figure 6: Example of moving out of radius.

original turning point.

- Change speed: Adjusts the movement speed of one path segment connected to a turning point to the next turning point.

4) Fitness Evaluation: Fitness evaluation in the proposed GA considers both the feasibility and route length. The feasibility of a route is determined by the two constraints, i.e., coverage and the within radius. Additionally, a rank-based assessment is employed.

- Coverage rate: If the route of the source device with the padding radius fails to cover the whole pattern, it is infeasible to perform the cutting process. Hence, this study utilizes the clipper offset [21] to examine the coverage by zooming in and out of the route via the padding radius. This method generates two contours to calculate the percentage of coverage for the pattern. If both contours fully cover the pattern, the coverage rate is set to 1; otherwise, it is infeasible and will not be ranked in fitness evaluation. Figure 5 illustrates an infeasible case with a coverage rate of 0.99, where the blue line represents the zooming-out of the route, and the green line represents the zooming-in of the route.
- Within-radius value: As the source device and the x-y axis motor operate simultaneously at different speeds, there is a risk that the current point with the padding radius of the source device may exceed the x-y axis motor's range, leading to the equipment aborting the processing procedure. To address this issue, we record the percentage that the source device safely traverses the pattern. If the route is within-radius, it is feasible and given 1; otherwise, the route is infeasible with a value smaller than 1. Figure 6 shows a processing moment, where the red line is the path of the source device, the black line is the pattern to be processed, and the circle represents the padding radius of the x-y axis motor. Nonetheless, infeasible solutions are still ranked because increasing the speed of the route may resolve this situation in the next generations.

Table I: Example of rank-based fitness evaluation

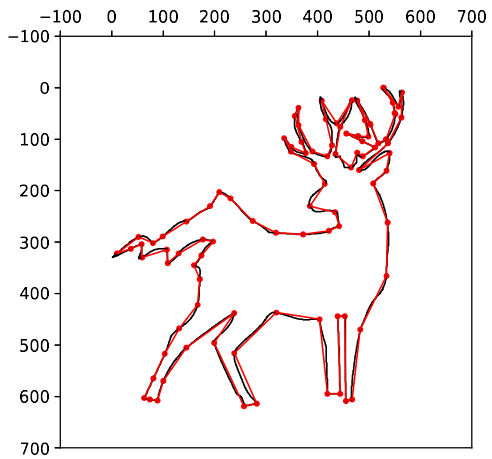
Rank	Coverage rate	Within-radius value	Route length (mm)
1	1	1	2340
2	1	1	2486
3	1	0.98	2384
4	1	0.95	2315
–	0.8	0.64	1876
–	0.7	0.53	1972

Table II: Parameter settings for GA.

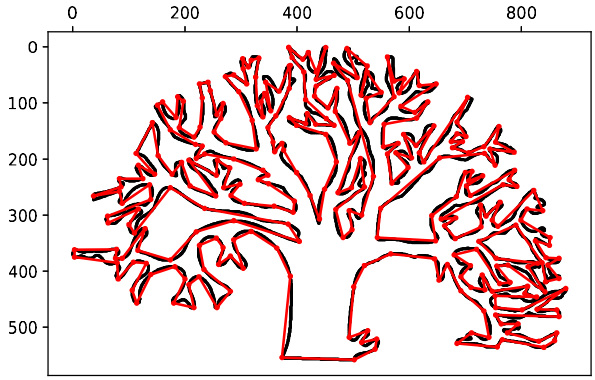
Parameter	Value
Representation	$\{\mathbb{R}, (\mathbb{N}, \mathbb{N})\}^{N_{\text{source}}}$
Population size	50
Parent selection	2-tournament
Crossover	Improved 2-point crossover
Crossover rate (p_c)	1.0
Mutation	Combination mutation
Mutation rate (p_m)	0.5
Survivor selection	$\mu + \lambda$
Termination	500 generations
#Runs	10 times

According to the above two considerations, the fitness value is defined by the Euclidean distance between points within the sum of cost in edge set E_s. The population is sorted primarily by the fitness value, measured by the route length, with solution feasibility taking precedence. On the other hand, the infeasible solutions that are out of radius but still in coverage are then sorted by the within-radius value. Table I provides an example of this rank-based fitness evaluation, where the solutions with ranks 1 and 2 are feasible and therefore sorted by route length. The solutions with ranks 3 and 4 are infeasible due to being out-of-radius but they are in coverage; thus, these solutions are sorted by the within-radius value. The last two solutions are not ranked because they are neither within-radius nor in coverage.

5) Survival selection: In this study, the survival selection follows the elitism $(\mu + \lambda)$ selection strategy, employing rank-



(a) Reindeer



(b) Tree

Figure 7: Test cases used in the experiments.

based selection, where the superior individuals progress to the next generation.

IV. Experimental Results

This study conducts several experiments to validate effectiveness and efficiency of the proposed method. Performance of the proposed method is examined in comparison to the route length gained from the LSA. The experiments adopt the sum of Euclidean distances between all LSA points V_{LSA} at initialization as a benchmark value for performance assessment. Two patterns of reindeer and tree with resolution $1mm \times 1mm$ are used as test cases. Figure 7a plots the pattern and the LSA points of the reindeer test case, which has 4385 points in the pattern with a total length 5127.05 mm and a benchmark value 4622.66 mm (97 points). Likewise, Fig. 7b shows the tree test case, which has 12803 points with a total length 15087.15 mm and benchmark value 13046.98 mm (314 points). Table II lists the parameter settings for the GA.

Table III and Figure 8 present the route lengths for the source device in both test cases. The results show that the proposed algorithm reduces the route length by 17.69–28.30%,

Table III: Average (Avg.), standard deviation (S.D.), and shortened percentage of source device's route lengths obtained from the GA on two test cases.

Test case	Route length (mm)		Shorten (%)
	Avg.	S.D.	
Reindeer	3314.23	35.66	28.30
Tree	10739.12	162.37	17.69

Table IV: Average (Avg.), standard deviation (S.D.), and decreased percentage of the sum of speed difference for the source device obtained from the GA on two test cases.

Test case	Sum of speed difference		Decrease (%)
	Avg.	S.D.	
Reindeer	36.65	3.59	69.46
Tree	278.47	9.99	47.42

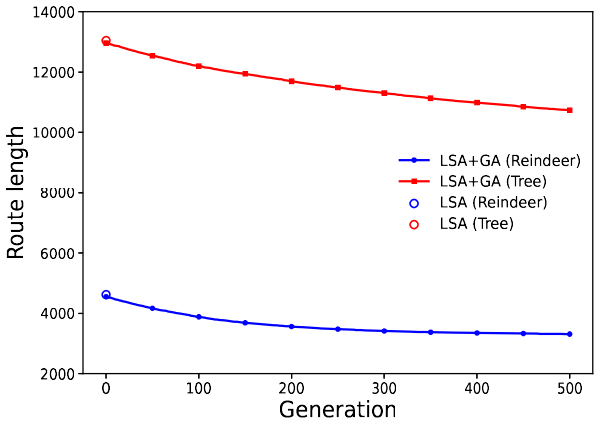


Figure 8: Progress of route length against number of generations for the source device over 10 runs.

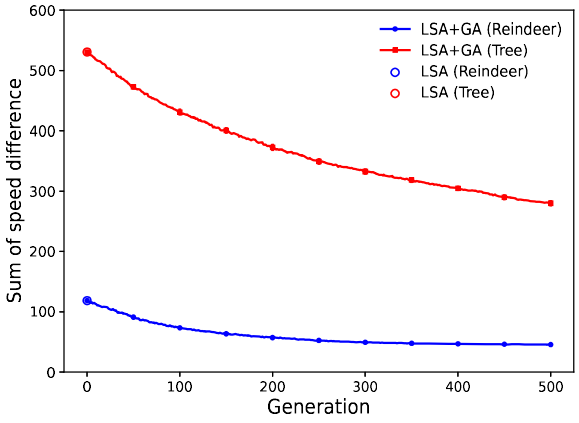


Figure 9: Progress of sum of speed difference against number of generations for the source device over 10 runs.

Table V: Average (Avg.), standard deviation (S.D.), and decreased percentage of source device's route lengths obtained from the GA on the reindeer test case.

Parameters	Sum of speed difference		Decrease (%)
	Avg.	S.D.	
p_c=0.5, p_m=0.9	4016.25	93.07	13.12
p_c=0.5, p_m=0.5	3614.94	72.99	21.80
p_c=0.5, p_m=0.1	3733.21	91.82	19.24
p_c=1.0, p_m=0.9	3484.40	70.99	24.62
p_c=1.0, p_m=0.5	3314.23	35.66	28.30
p_c=1.0, p_m=0.1	3632.70	74.93	21.42

Table VI: Average (Avg.), standard deviation (S.D.), and decreased percentage of source device's route lengths obtained from the GA on the tree test case.

Parameters	Sum of speed difference		Decrease (%)
	Avg.	S.D.	
p_c=0.5, p_m=0.9	11991.19	133.77	8.09
p_c=0.5, p_m=0.5	11640.25	60.18	10.78
p_c=0.5, p_m=0.1	11923.87	149.79	8.61
p_c=1.0, p_m=0.9	11360.89	130.85	12.92
p_c=1.0, p_m=0.5	10739.12	162.37	17.69
p_c=1.0, p_m=0.1	11582.48	209.14	11.22

showing its effectiveness in optimizing routes. This reduction in the overall route length of the source device can contribute to an improved speed for the cutting process.

Table IV and Figure 9 showcase the results of the sum of speed differences for the source device. While the sum of speed difference is not an objective in this routing problem, the proposed algorithm can still reduce the sum of speed difference by 47.42–69.46% compared to the routes constructed only by LSA points. This reduction in the sum of speed differences contributes to a decreased number of acceleration and deceleration of the source device, leading to a smoother cutting process and improved processing quality.

Table V and Figure 10 show the effects of parameters on route length in the reindeer test case, while Table VI and Figure 11 demonstrate these effects in the tree test case. The results indicate that a higher crossover rate p_c and a medium mutation rate p_m lead to a significant reduction in route length in both test cases. This trend may be attributed to the exploitation of more solutions with diverse route combinations enabled by a higher crossover rate. Additionally, a higher mutation rate was not selected due to the increased likelihood of introducing infeasible or lower-quality solutions, while a lower mutation rate would be insufficient to maintain diversity.

Figure 12 displays the best result for the reindeer test case, with a route length of 2863.07 mm. Similarly, Figure 13 presents the best result for the tree test case, with a route length of 9833.02 mm. The results show that the source device moves seamlessly through the complicated sections of the pattern and traverses the simpler parts at a higher movement speed, which achieves an ideal route for the cutting process.

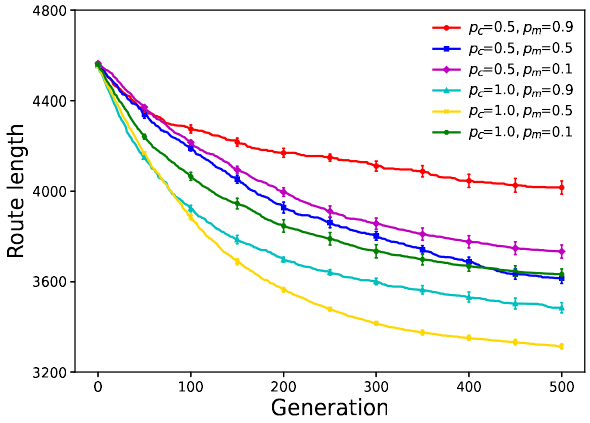


Figure 10: Progress of route length against number of generations for the source device over 10 runs on the reindeer test case.

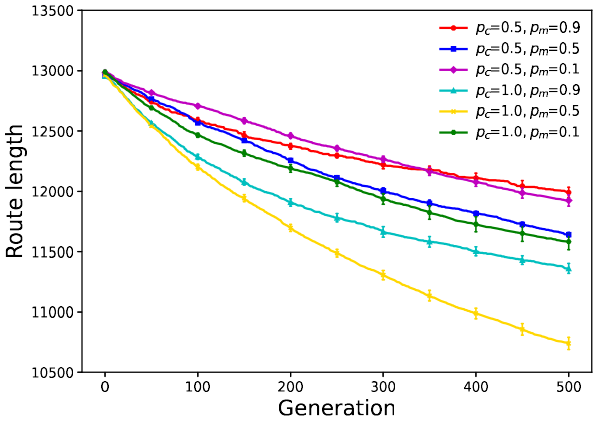


Figure 11: Progress of route length against number of generations for the source device over 10 runs on the tree test case.

V. Conclusions

This study presents a novel way to address the route optimization problem for multi-axis motion control, which is pivotal to numerous applications in the manufacturing industry. Specifically, we formulated the routing problem as the CETSP and, moreover, designed a GA to deal with this problem. Differing from the existing studies on the CETSP, the permutation of city points is predetermined using two LSA approaches, i.e., RDP and VW algorithms. This study further presents new crossover and mutation operators to improve the GA. The effectiveness of the proposed approach is examined through two test cases. The experimental results indicate that the GA achieves 28.30% and 17.69% reduction in route length on these two test cases. In addition, it decreases the sum of speed differences by 69.46% and 47.42%. These satisfactory outcomes validate the effectiveness and efficiently of the proposed GA in reducing route length and smoothing the speed variation for the source device, thereby enhancing the quality

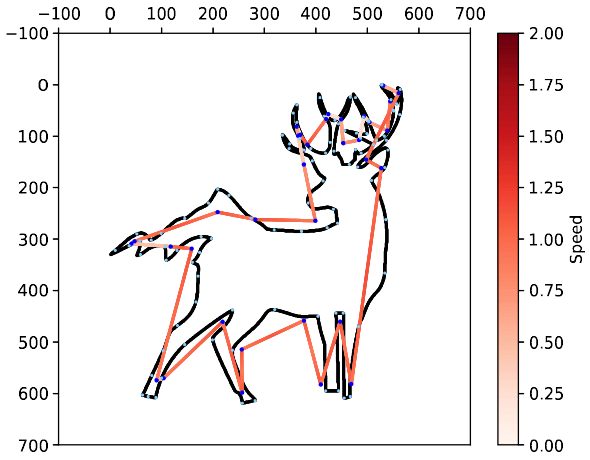


Figure 12: Best route obtained for the reindeer test case.

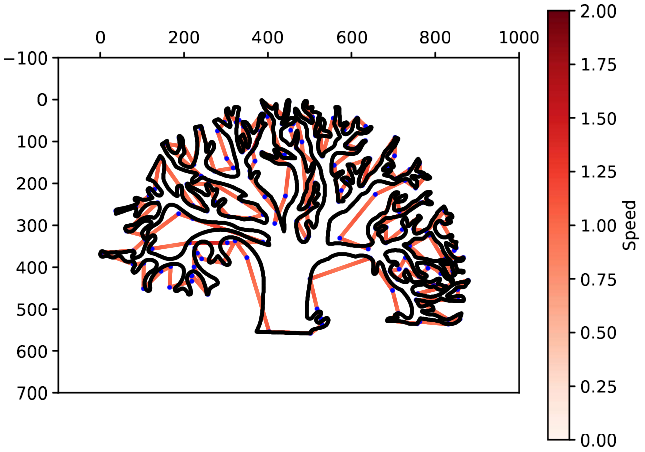


Figure 13: Best route obtained for the tree test case.

of manufacturing process. Some directions remain for future study. The experiments can be expanded by including multiple cutting patterns. Integration of local search or heuristics in the GA is also promising for enhancing the results of route planning for multi-axis motion control.

References

[1] A. Tavaeva, A. Petunin, S. Ukolov, and V. Krotov, *A Cost Minimizing at Laser Cutting of Sheet Parts on CNC Machines*. Springer International Publishing, 2019, pp. 422–437.

[2] M. Cui, L. Lu, Z. Zhang, and Y. Guan, "A laser scanner-stage synchronized system supporting the large-area precision polishing of additive-manufactured metallic surfaces," *Engineering*, vol. 7, no. 12, pp. 1732–1740, 2021.

[3] W. Serruys, *Sheet metalworking: State of the art*. LVD Company, 2008.

[4] H. Yeung, J. Neira, B. Lane, J. Fox, and F. Lopez, "Laser path planning and power control strategies for powder bed fusion systems," in *Proceedings of 2016 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2016.

[5] A. Petunin, "General model of tool path problem for the CNC sheet cutting machines," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2662–2667, 2019.

[6] R. A. Faizrakhmanov, R. T. Murzakaev, A. V. Burylov, and V. S. Shilov, "Formation of an energy-efficient cutting tool path in waterjet and laser cutting CNC machines," *Russian Electrical Engineering*, vol. 86, no. 11, pp. 651–655, 2015.

[7] 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.

[8] M. J. Bermingham, P. Schaffarzyk, S. Palanisamy, and M. S. Dargusch, "Laser-assisted milling strategies with different cutting tool paths," *The International Journal of Advanced Manufacturing Technology*, vol. 74, pp. 1487–1494, 2014.

[9] D. J. Gulczynski, J. W. Heath, and C. C. Price, "The close enough traveling salesman problem: A discussion of several heuristics," in *Operations Research/Computer Science Interfaces Series*. Springer US, 2006, pp. 271–283.

[10] F. Carrabs, C. Cerrone, R. Cerulli, and M. Gaudioso, "A novel discretization scheme for the close enough traveling salesman problem," *Computers & Operations Research*, vol. 78, pp. 163–171, 2017.

[11] B. Behdani and J. C. Smith, "An integer-programming-based approach to the close-enough traveling salesman problem," *INFORMS Journal on Computing*, vol. 26, no. 3, pp. 415–432, 2014.

[12] W. P. Coutinho, R. Q. do Nascimento, A. A. Pessoa, and A. Subramanian, "A branch-and-bound algorithm for the close-enough traveling salesman problem," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 752–765, 2016.

[13] X. Wang, B. Golden, and E. Wasil, "A steiner zone variable neighborhood search heuristic for the close-enough traveling salesman problem," *Computers & Operations Research*, vol. 101, pp. 200–219, Jan. 2019.

[14] W. K. Mennell, B. L. Golden, and E. Wasil, "A steiner-zone heuristic for solving the close-enough traveling salesman problem," in *Proceedings of 12th INFORMS Computing Society Conference*, ser. ICS 2011. INFORMS, 2011.

[15] W. K. Mennell, "Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, and sequence-dependent team orienteering problem," Ph.D. dissertation, University of Maryland, College Park, 2009.

[16] A. D. Placido, C. Archetti, and C. Cerrone, "A genetic algorithm for the close-enough traveling salesman problem with application to solar panels diagnostic reconnaissance," *Computers & Operations Research*, vol. 145, p. 105831, 2022.

[17] A. D. Placido, C. Archetti, C. Cerrone, and B. Golden, "The generalized close enough traveling salesman problem," *European Journal of Operational Research*, vol. 310, no. 3, pp. 974–991, 2023.

[18] R. B. MCMASTER, "Automated line generalization," *International Journal for Geographic Information and Geovisualization*, vol. 24, no. 2, pp. 74–111, 1987.

[19] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer Graphics and Image Processing*, vol. 1, no. 3, pp. 244–256, 1972.

[20] M. Visvalingam and J. D. Whyatt, "Line generalisation by repeated elimination of the smallest area," University of Hull, Tech. Rep., 1992.

[21] X. Chen and S. McMains, "Polygon offsetting by computing winding numbers," in *Proceedings of Design Automation Conference*, 2005.