

A NOVEL METHOD FOR ACHIEVING MINIMUM DISTANCE COLLISION-FREE TOOL PATH FOR DRILLING

Z. Khodabakhshi¹, A. Hosseini^{1*}, A. Ghandehariun²,

¹Department of Automotive, Mechanical, and Manufacturing Engineering, Faculty of Engineering and Applied Science, Ontario Tech University, Oshawa, Canada

²Mechanical Engineering Department, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

*sayyedali.hosseini@uoit.ca

Abstract - In recent years, many researchers have studied drilling tool path generation. Tool travelling time is a major portion of overall manufacturing time particularly in large scale items to be drilled. Nowadays, tool path is selected from a set of regular tool paths offered in CAM systems. These tool paths are mostly far from being the optimum path. In this paper, a new Travelling Salesman Problem (TSP) is utilized for generating a safe tool path in drilling processes. The developed optimization model can handle the objective function under multiple constraints, namely, returning to the tool origin rather than the initial hole to be drilled, considering all workpiece dimensions/corners for the initial position of the drill bit, as well as an obstacle constraint to generate a collision free trajectory. Performance of the proposed model is obtained for different initial tool positions on all corners of the workpiece. The results prove that the model can solve the problem in a timely manner and is able to generate an optimum safe path for tool traverse.

Keywords; *Tool path optimization, TSP model, Collision free drilling path, the Nearest Neighborhood method.*

I. INTRODUCTION

Over the past decades manufacturing industry has faced rapid growth in its processes. Manufacturing processes are restricted by their scope and activity time. Since an appropriate motion path is highly effective on the manufacturing process time, many studies have been performed to achieve an optimum motion path. A motion path defines a continuous movement of tool relative to workpiece and can be either point to point as in drilling or continuous as in milling.

Many researchers have used the Travelling Salesman Problem (TSP) as an efficient strategy for sequencing problems in the field of manufacturing. As reported by Dewil et al. [1], 79% of studies applied the TSP model to solve the optimum path in drilling. The aim of the TSP model is to minimize the length of the movement path between points, which may be holes or welds [2].

The current research tries to apply TSP algorithm for optimizing the tool path in drilling. A novel practice is proposed and introduced. This method involves adding new constraints, which were mostly ignored in similar studies which used the TSP model for generation and optimization of a safe point-to-point drilling path. This paper considers the drilling process; however, the logic of the presented model can also be applied to optimization of continuous tool paths, such as milling tool path.

The structure of this paper is as follows. In the following section a detailed overview of current approaches for tool path optimization is presented where the gap between real world industry problems and academia is identified. Then a description of the proposed optimization model used for formulating the problem is introduced. Finally, a test case is demonstrated for assessing the approach. Results are discussed and analyzed, and the main conclusions are drawn.

II. LITERATURE REVIEW

Minimizing the manufacturing time is considered to be one of the important factors in increasing the rate of production of goods. TSP is a popular method in optimization. A salesman in TSP must visit N cities, with the assumption that each city must be visited exactly once and salesman must return to the starting city. In order to cut down manufacturing time, finding the optimum path can be corresponded to the TSP problem [3]. Although applications in logistics and transportation are the most common for the TSP, the flexibility and effectiveness of the model have led to many applications in other areas. In a drilling machine, scheduling holes to be drilled is a classic case for the TSP application [4].

Onwubolu [3] employed the TSP method with Differential Evolution (DE) for CNC drilling operations. DE was solved using both forward and backward transformation techniques. The technique is applied for drilling a Printed Circuit Board (PCB), which is a flat laminated polymer composite. The comparison between DE and other heuristic optimization techniques showed that the path generated by TSP/DE performed better than other methods.

Huizar et al. [5] solved the TSP problem using Artificial Immune System (AIS). Clonal Selection Algorithm (CSA) as a common class of algorithms in AIS was used to decrease the drilling time and cost by generating the optimal sequence of G-codes. Three different experiments were done. A comparison with the G-code generated by the CAD/CAM software showed that the CSA generated a significantly shorter path for drilling; the manufacturing time was reduced by 35% to 53%.

Other heuristic models have also been applied to optimization of the TSP path such as Ant Colony, Simulated Annealing, Genetic algorithm and Particle Swarm Optimization [4]. The problem with the heuristic algorithms, which have been mainly used in recent years, is that the initial selection of the cities is random. To overcome this problem, recent studies suggest the application of hybrid models for initialization method [4, 6]. The hybrid technique selects the initial path by classic TSP algorithms, such as the nearest neighborhood, greedy algorithms, and local search approaches, and tries to further optimize the initial path by heuristic algorithms [7].

Oysu and Z. Bingul [7] used a hybrid genetic algorithm and the simulated annealing model to minimize the air time during milling. The results indicated that GA generated a good selection of starting points for running SA and thus, the hybrid algorithm generates better results compared to each of the models alone.

To improve the application of the TSP method in manufacturing fields, it is crucial to deeply consider the basics of TSP algorithms and change/reconsider its assumptions if needed to particularly meet manufacturing needs in the real world. In most of the drilling problems there is no need to return to the first drilled hole/city. However, the studies which applied TSP to the drilling process all consider a closed loop for the tool path. Connecting the final city to the initial city adds an extra distance to the path which is not required in actual drilling practice. This both adds extra distance to the path as well as extra non-value-added time to the drilling process overall time. In actual situations, the tool starts from a predefined origin and travels through all the holes and returns to the origin position. In recent studies, the tool is assumed to return to the tool origin at the end of the process [5]. A possible technique is to solve the TSP problem with the closed loop assumption and then exclude the last distance from the final travelled distance. Onwubolu and Clerc [4] studied a TSP problem with 9 cities and proposed a PSO algorithm for drilling. According to the results obtained from the PSO model the optimum tour was 1, 4, 7, 9, 8, 5, 6, 3, 2, 1 and the travelled distance was 2000 mm. For comparison purposes, they excluded the distance from the last hole to the starting city, and thus, the path and the travelled distance changed to 1, 4, 7, 9, 8, 5, 6, 3, 2 and 1800 mm, respectively. Using this obtained path on a real drilling machine, the drill bit would visit city 1 after drilling city/hole number 2 and start drilling an existing hole and then return to the safe initial position. This will add extra time to the drilling total time.

Another factor to consider in comparison of the reviewed algorithms is while optimization with complicated heuristic models leads to optimized tool paths, minimizing the model computational time is also important. A hybrid GA/SA model was used in the work of Skinderowicz [8] for 200 to 500 nodes with a computational time of 600 seconds. This computational

time is rather high for real world application. The long computational time is reported as one of the drawbacks of hybrid techniques [7].

Generating a safe tool path and avoiding collisions between the machine tool and the workpiece is a major problem for high speed multi-axis machines. Collisions can cause damages to the machine or the workpiece which will add additional cost to manufacturing [9]. Detecting the possibility of collisions and avoiding them have many applications in industry. Modern machines have sensors which are able to detect the collision, however, they leave the decision to the human operator and will stop the operation [10]. Optimization techniques have been widely used for generating safe trajectories for manipulators in the robotics industry. Lee and Kim [11] applied genetic algorithm to plan an intelligence path for robots. They also used an acyclic graph for the most effective initial path sets.

In this study the gaps in the available literature are addressed through presenting a new TSP model with specified constraints for drilling processes. The Nearest Neighborhood (NN) solution for the TSP model is used for achieving this objective.

III. PROPOSED MODEL

The popular Traveling Salesman Problem (TSP) is a famous method in optimization techniques. TSP is one of the best-known studies of optimization problems and is classified as NP (non-deterministic polynomial time)-hard problems [12]. It means that finding the exact optimal solution for TSP will be suffering from exponentially growing complexity and thus, algorithms will help find near-optimal solutions [13]. Finding exact solutions are difficult even for simple NP-hard problems [2]. In TSP, a salesman has to visit n cities. Each city must be visited only once and the salesman must return to the home city. The TSP goal is to find the minimum total distance that the salesman travels. The initial definition of the TSP creates some subtours. These subtours satisfy the assumption of visiting all the cities and returning to the first city of the subtour. Since having a subtour is not accepted in TSP, many mathematical formulations are used to disallow subtours in the TSP problem [7]. The frequent one is to let K_{ij} be a decision variable for the purpose of no looping constraints with the following definition:

$$K_{ij} = 0 \text{ or } 1 \quad (1)$$

Let K_{ij} be 1 if city j is visited immediately after city i and 0 otherwise. TSP is then defined as minimizing the following summation:

$$\sum_{i=1}^n \sum_{j=1}^n d_{ij} K_{ij} \quad (2)$$

Subject to

$$\sum_{i=1}^n k_{ij} = 1, \forall j \quad (3)$$

$$\sum_{j=1}^n k_{ij} = 1, \forall i \quad (4)$$

where n is the number of cities and d_{ij} refers to the distance from city i to j . TSP can be categorized into two approaches: symmetric TSP and asymmetric TSP. If the distances between the cities differ depending on the movement direction, formulation is asymmetric, and otherwise it is symmetric. The set of constraints in (3) ensures that the tool visits each hole and constraints in (4) ensure that the tool leaves every hole [1].

According to Oysu and Bingul [7], individual looping is not accepted in the original problem. There is no pre-defined priority on any cities and there is no need to return or visit a city prior to all other cities, so we do not have any sub tours[3]. In drilling process, each node is determined by x, y coordinates, which can be used for calculating the Euclidean distance between cities i and j (Euclidean TSP).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

A. Proposed model description

In this study, the TSP assumptions have been reconsidered so that the tool returns to a predefined origin position instead of the starting city. This study considers the four corners of the workpiece as possible tool initial positions to find out which position creates the lowest overall travel distance based on the cities' coordinates.

The nearest neighborhood algorithm starts by selecting a random city. The algorithm proceeds through $n - 1$ stages, in each stage adding an unassigned city to the loop that is closest to the current city. Then the algorithm investigates whether the path to the next city has a collision with the obstacle. The sequence progresses by all remaining cities at each stage to meet all the constraints. The Algorithm will be repeated with each time an initial selection of a different city. Finally, the optimum path will be selected. The whole process will be performed for all workpiece corners to achieve the minimum path traversed by the tool. In this research, only 2D movement of the tool is considered. The computational steps for the application of the proposed model are defined as follows:

Step 1. Initialize from one/each corner of the workpiece and a start city. For $i = 1, 2, \dots, \text{the number of cities } (n)$

Step 2. Select a new city from the matrix, which establishes the minimum distance according to (2). for $j = 1, 2, \dots, n - 1$.

Step 3. Investigate whether the path to the new city has a collision with the obstacle or not.

If "Yes" the tool will proceed to the nearest obstacle edge to avoid any collision and then select the next nearest city. If "No" the algorithm will move toward the next city.

Step 4. Update unassigned cities.

Algorithm will run until all the cities are visited.

Step 5. Algorithm will consider a path to the tool origin considering the fact that there should not be a collision on the path from the tool origin to the start city as well as the path from the last city to the tool origin.

Step 6. The least overall travelling distance will be selected in each corner.

Step 7. For comparison purposes, distances from step 6 will be compared and the final path will be selected.

As can be seen, the nearest neighborhood is a constructive method. It is worth to mention that in constructive methods, the solution is found by adding components to a partial solution until the final solution is achieved. According to Onwubolu [3], the workload will be too large when the number of cities, n , increases. As a solution, the starting city may be selected randomly instead of selecting all the cities. Random selections of the initial city, as discussed in section II, can cause problems and thus, hybrid models are preferred [6].

B. A case study

In order to understand the flow of the proposed algorithm, a simple case is studied. It is assumed that the motion command is executed on a computer numerically controlled (CNC) machine to drill holes on a simple workpiece. The depth and diameter of holes are considered consistent among the holes. Therefore, the z travel distance for creating the holes is similar among the holes and can be eliminated from the calculation. Likewise, no tool change is required during the process. As a result, the motion is considered 2D in the x and y directions. The drilling path can be pictured as a TSP, where the salesman is drill-bit and the locations of holes are cities. Figure 1 shows the dimensions of the workpiece with four corners (adjacent to the four corners of figure), three holes to be drilled (cities to be travelled), and an obstacle (depicted as a red line). The obstacle on the middle of the workpiece forces the tool to move around it to avoid any collision.

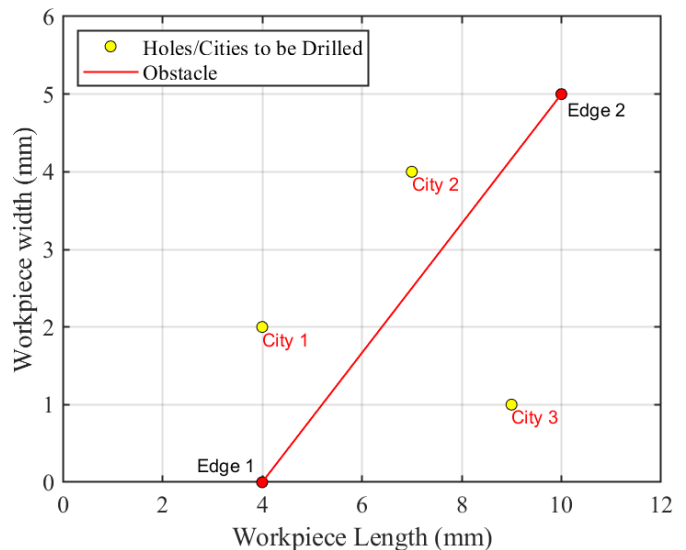


Figure 1. Workpiece dimensions, edges and the cities

Table I and Table II show the x, y coordinates of holes (cities), obstacle edges, as well as workpiece corners. Regarding the metric classification for distance, the Euclidean movement, which is the straight line between two cities, is considered in Euclidian TSP. Using (5), the distance matrix is constructed as an $n * n$ matrix of the cities, as shown below in (6):

$$\begin{bmatrix} 0 & 3.6 & 5.09 \\ 3.6 & 0 & 3.6 \\ 5.09 & 3.6 & 0 \end{bmatrix} \quad (6)$$

MATLAB R2018a is used to run the algorithm. For the problem presented in Fig. 1, with the tool origin located at (0, 0), the optimum TSP tour is tool origin (0, 0), city 1, city 2, edge 2, city 3, edge 1, and tool origin, see Fig. 2. For such a tour (considered as the tool path), the objective function value or the path with shortest distance is 24.46 mm. The total run time is 0.118 seconds. Needless to mention that dimensions in mm are just for verification of the proposed method. The method can be implemented to different geometries with any other dimensions.

TABLE I. CITIES AND EDGES COORDINATES

(x,y)	Holes (Cities) and Edges					
	City 1	City 2	City 3	City 4	Edge 1	Edge 2
x	4	7	9	4	10	4
y	2	4	1	0	5	2

TABLE II. WORKPIECE CORNER COORDINATES

(x,y)	Corners			
	Corner 1	Corner 2	corner 3	corner 4
x	0	12	12	0
y	0	0	6	6

The algorithm can be executed again with the bottom right corner (12, 0) as the tool origin. The optimum TSP tour for such a configuration is tool origin (12, 0), edge 2, city 2, city 1, edge 1, city 3, and tool origin, see Fig. 3. The shortest distance for this path is 22.41 mm. The total run time is 0.107 seconds.

The same steps are followed for the remaining corners (Fig. 4 and Fig. 5). For ease of referencing and naming the optimum tool path, edge 1 is indexed as 4, edge 2 is indexed as 5, and tool origin is indexed as 0. The results are shown in table III. As can be seen in table III, the optimum path will be 0, 2, 1, 4, 3, and 0 with the shortest path length of 21.92 mm and 0.105 seconds of computation time. For practical applications this time is acceptable. Thus, the operator can define (12, 6) as the safe initial position for the tool.

To further analyze the performance of algorithm in terms of computational time, examples with 40, 80 and 110 cities on an imaginary workpiece of 100 mm × 50 mm are studied. Cities and edges are selected randomly within the scales. Table IV shows the final runtime results. As can be seen, increasing the number of cities significantly increases the complexity of the problem and leads to an increase in the computational time, but yet within an acceptable range. The contributions of the proposed algorithm can be summarized as follows:

- The tool origin is included in the TSP model
- Obstacles are included to increase the applicability of the mathematical basis in generating a collision-free tool motion path

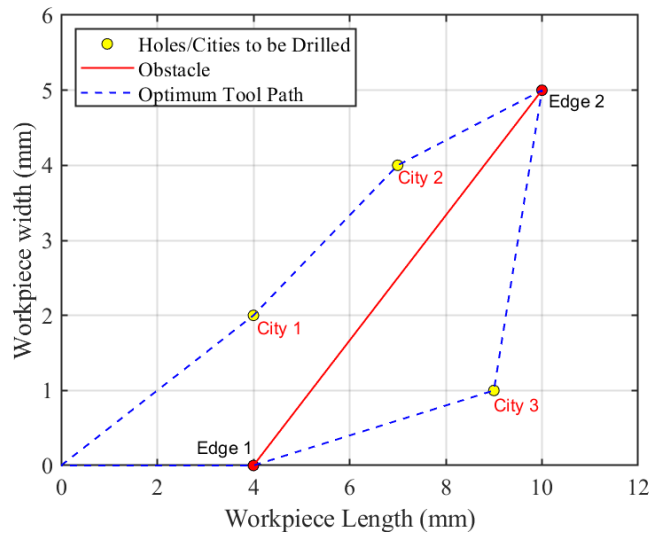


Figure 2. Shortest Path, tool origin (0, 0)

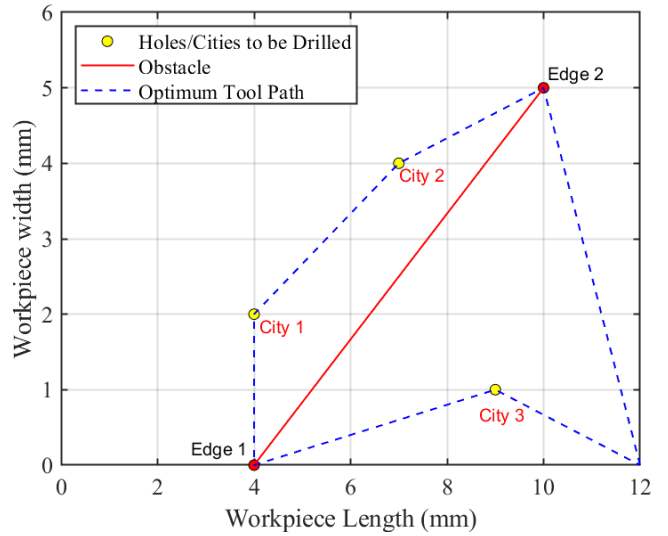


Figure 3. Shortest path, tool origin (12, 0)

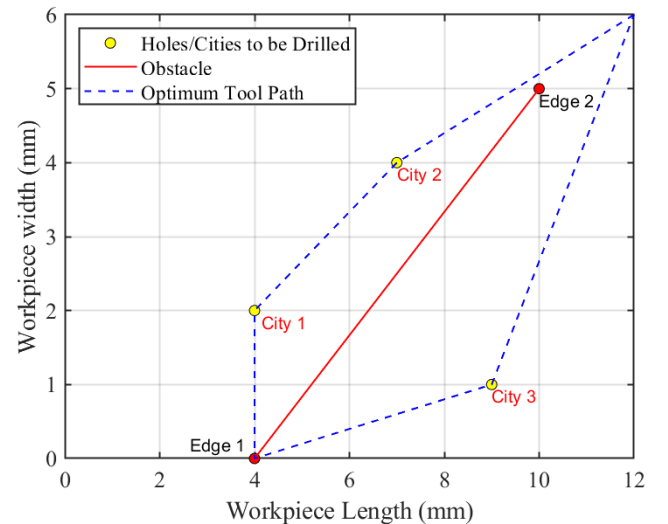


Figure 4. Shortest path, tool origin (12, 6)

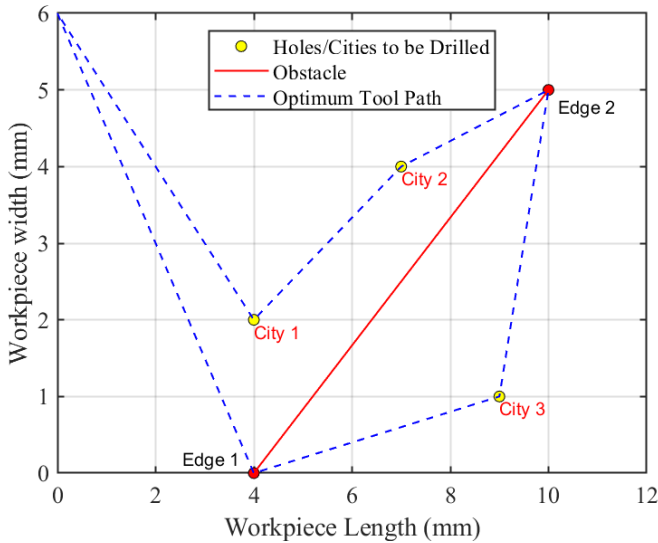


Figure 5. Shortest path, tool origin (0, 6)

TABLE III. COMPARISON OF ALGORITHM RESULTS FOR VARIOUS STARTING POINTS

Starting point	Algorithm results		
	Shortest path	Shortest path length (mm)	Run time (seconds)
Corner 1	0,1,2,5,3,4,0	24.46	0.118
Corner 2	0,5,2,1,4,3,0	22.41	0.107
Corner 3	0,2,1,4,3,0	21.92	0.105
Corner 4	0,4,3,5,2,1,0	28.86	0.110

TABLE IV. ALGORITHM RUNTIMES FOR LARGER NUMBER OF CITIES

	Number of cities			
	3	40	80	110
Runtimes (s)	0.105	0.543	1.885	2.862

IV. CONCLUSION

The contributions presented in this paper mainly focus on the mathematical viewpoint of the TSP model and try to address the most recent problems in industry. In this paper, a model is proposed for generation of an optimized and safe tool path for CNC drilling. By adding new constraints to the TSP model and considering possible obstacles on the tool path, the developed algorithm achieved an optimized and collision-free tool path for point-to-point drilling. The application of presented algorithm in several case studies proved the performance of the algorithm within an acceptable computation time.

V. FUTURE WORKS

TSP solved by the nearest neighborhood method is one of the most studied topics in the sequencing problems with a considerable portion of 79% [1]. In recent years, hybrid heuristic models emerge as having great potential in optimization techniques. Further research can include applying the proposed

constraints in hybrid models. The proposed algorithm can be employed in 3D CNC machines as well as milling processes.

REFERENCES

- [1] R. Dewil, İ. Küçüköğlü, C. Luteyn, and D. Cattrysse, "A critical review of multi-hole drilling path optimization," *Archives of Computational Methods in Engineering*, vol. 26, no. 2, pp. 449-459, 2019.
- [2] A. Nassehi, W. Essink, and J. Barclay, "Evolutionary algorithms for generation and optimization of tool paths," *CIRP Annals*, vol. 64, no. 1, pp. 455-458, 2015.
- [3] G. C. Onwubolu, "Optimizing CNC drilling machine operations: traveling salesman problem-differential evolution approach," in *New optimization techniques in engineering*: Springer, 2004, pp. 537-565.
- [4] G. Onwubolu and M. Clerc, "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization," *International Journal of Production Research*, vol. 42, no. 3, pp. 473-491, 2004.
- [5] C. Huizar, O. Montiel-Ross, R. Sepúlveda, and F. J. D. Delgadillo, "Path planning using clonal selection algorithm," in *Recent Advances on Hybrid Intelligent Systems*: Springer, 2013, pp. 303-312.
- [6] A. K. Gupta, P. Chandna, and P. Tandon, "Hybrid genetic algorithm for minimizing non productive machining time during 2.5 D milling," *International Journal of Engineering, Science and Technology*, vol. 3, no. 1, 2011.
- [7] C. Oysu and Z. Bingul, "Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 3, pp. 389-396, 2009.
- [8] R. Skinderowicz, "An improved ant colony system for the sequential ordering problem," *Computers & Operations Research*, vol. 86, pp. 1-17, 2017.
- [9] T. Moriwaki, "Multi-functional machine tool," *CIRP Annals*, vol. 57, no. 2, pp. 736-749, 2008.
- [10] R. Ahmad and P. Plapper, "Generation of safe tool-path for 2.5 D milling/drilling machine-tool using 3D ToF sensor," *CIRP Journal of Manufacturing Science and Technology*, vol. 10, pp. 84-91, 2015.
- [11] J. Lee and D.-W. Kim, "An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph," *Information Sciences*, vol. 332, pp. 1-18, 2016.
- [12] D. Pezer, "Efficiency of tool path optimization using genetic algorithm in relation to the optimization achieved with the CAM software," *Procedia Engineering*, vol. 149, pp. 374-379, 2016.
- [13] J. Gu and X. Huang, "Efficient local search with search space smoothing: A case study of the traveling salesman problem (TSP)," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 5, pp. 728-735, 1994.