






Efficient Toolpath Planning for Voxel-Based CNC Rough Machining

Aman Kukreja¹ , Mandeep Dhanda²  and S. S. Pande³ 

¹Indian Institute of Technology Bombay, amankukreja@iitb.ac.in

²Indian Institute of Technology Bombay, mandeepdhanda@iitb.ac.in

³Indian Institute of Technology Bombay, s.s.pande@iitb.ac.in

Corresponding author: S. S. Pande, s.s.pande@iitb.ac.in

Abstract. Multi-axis CNC machines are widely used to manufacture complex industrial parts such as dies and molds. The current CNC toolpath planning strategies are complex and often lead to inefficient part programs. This paper presents an approach to use voxel-based CAD models for efficient zig-zag toolpath planning. The developed system focusses on the rough milling of complex parts having multiple machining features. The system takes the CAD (STL) part model and identifies the machinable and non-machinable areas by analyzing the voxelized part model. This follows the segmentation of machinable area into smaller machining regions. Genetic Algorithm is then used to generate an optimum sequence of machining these regions to reduce air cutting path. The developed system was extensively tested using parts with varying machining feature complexities. The system was found to be better than the traditional zigzag roughing toolpath generation methods in terms of the reduction of the number of tool liftoffs and air cutting path length.

Keywords: CNC rough machining, Voxel-based CAD model, GA based toolpath planning.

DOI: <https://doi.org/10.14733/cadaps.2021.285-296>

1 INTRODUCTION

Today, multi-axis Computer Numerical Control (CNC) machines are widely used to manufacture complex industrial products in automobile, aerospace, and die molds industries. The availability of efficient part programs directly governs the productivity and part quality during CNC machining. The process planners use the Computer-Aided Manufacturing (CAM) systems, which often provide inefficient toolpath. In addition, they need several user inputs and iterations, which are cumbersome and time-consuming. A need thus exists to develop toolpath planning algorithms to generate efficient CNC part programs.

The roughing stage of CNC machining takes a significant portion of the total machining cycle time during the manufacture of a part. Roughing strategy aims to achieve near net shape and minimize the total cycle time. Literature documents the development of several toolpath planning

strategies for multi-axis (3-5 axis) CNC machining. Various studies on 2.5 D rough milling using toolpath patterns such as zig-zag, spiral, and contour-based have been reported [29]. Kim and Choi [13] observed that smooth zig-zag toolpath gives less machining time than the contour parallel. Generally, in practice, the zig-zag toolpath pattern is most widely used for die and mold machining because of the enhanced constant cutting loads [17].

To improve productivity (minimize cycle time) for large parts with several machining features, there are two main requirements viz. a smaller number of tool lift offs and the optimized sequence of operations. Several studies report work in this direction for applications such as drilling [1][26], laser cutting [10], yarn cutting [16], and milling segment [12]. Evolutionary optimization techniques have been reported to reduce air cutting time by considering path planning as a traveling salesman problem (TSP). Castelino et al. [6] used the LK heuristic method while Oysu and Bingul [23] used Simulated Annealing (SA) and Genetic Algorithm (GA) methods to find an optimal way of connecting (stitching) milling sub-regions. They fixed the start and end of the sub-regions apriori, which made the solution sub-optimal. Pavanaskar et al. [24] used a greedy approach for joining geometry streamlines to generate an energy-efficient toolpath for pocket milling. Barclay et al. [5] used a grid-based strategy for simple prismatic parts to find an optimal toolpath using genetic programming. Nassehi et al. [21] reported the use of the STEP-NC model with GA to generate an optimized toolpath. The approach was, however, unable to give better performance than the commercially available software.

All the above strategies work on CAD models in the B-Rep, STL, or parametric equation form to generate toolpath and often need feature recognition which require complex geometric computations. Researchers are thus, exploring the use of voxel-based CAD models for toolpath planning. The voxel representation shifts the computation from real variables to the integer and Boolean space. It specifies the cutter locations by integer indexes that are generated by checking whether the traversing voxel is filled or empty.

Literature reports scant research work on the use of voxel-based CAD models for CNC rough toolpath planning. Jang et al. [11] used the voxel-based model to simplify the computation of regularized Boolean set operations for showing the in-process state of the workpiece. Tarbutton et al. [27] used a ray casting approach to calculate cutter contact (CC) points through GPU (Graphics processing unit) based parallel processing for the generation of roughing toolpath. Tarbutton et al. [28] further extended their work to generate a gouge free roughing and finishing toolpath. Konobrytskyi et al. [15] described a way to determine the cutter contact point for 5-axis machining using GPGPU. However, their algorithm required complex computational tasks such as offset surface calculations. The Iso-scallop toolpath generation using voxel model was reported by Balabokhin and Tarbutton [3][4]. A voxel-based surface offsetting method was proposed by Lynn et al. [19] for a virtualized voxel-based CAM package. Lynn et al. [20] proposed an alternative to additive manufacturing using voxel-based CNC machining for non-assembly mechanisms. The work was, however, part specific. Shen and Tarbutton [25] introduced a voxel-based toolpath planning approach for welded and cast near net shape parts using the scanned data as the stock material.

Literature shows that there is a need for the development of a comprehensive methodology using the voxel-based CAD models for efficient CNC roughing toolpath planning. The present work is an attempt in this direction.

2 METHODOLOGY

2.1 Modular Diagram

Figure 1 shows the functional modules of the developed system for the rough machining operation on 3-axis CNC milling machine. The system takes the STL CAD part model as input and converts it into the voxel part model. The voxel data, thus generated, is preprocessed to provide stock allowance (for finishing) and fill in the inaccessible (blind) feature for 3-axis CNC machining. The

modified voxel model data is then parsed to generate toolpath. Post-processing of the tool path data is carried out to add the controller-specific instruction to carry out real machining. Various functional modules of the developed system are explained one by one.

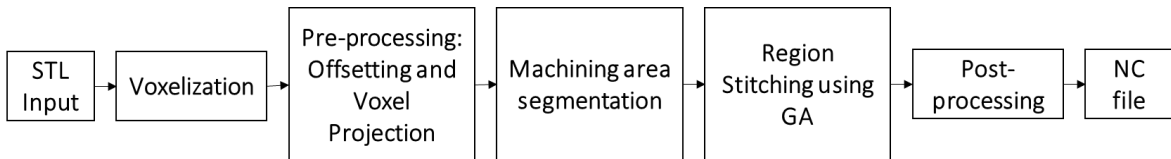


Figure 1: Modular diagram.

2.2 Voxelization

Voxelization is a method to convert an object model into a stack of cuboidal blocks (voxels) that best approximate the shape. The process of voxelization takes the STL CAD part model as an input and compute the extents of the cuboidal raw stock. The part model is then voxelized to form machinable ('0'), and non-machinable voxels ('1') using the ray-tracing algorithm [13]. The voxel size is limited by the tool diameter to prevent undercuts. The size of the voxel (Δx , Δy , Δz) is taken equal to the user-defined values of the step over (Δx , Δy) and the depth of cut (Δz), respectively.

In the present work, the voxel-based part model is represented as a 3D array. The index (i , j , k) of each element in the array corresponds to the position of the voxel. When multiplied by the size of the voxel ($x=i\times\Delta x$; $y=j\times\Delta y$; $z=k\times\Delta z$), it gives the position of the voxel in the Cartesian space. The value stored in the data structure array is in the binary form, i.e., either '1' (material is present) or '0' (material is absent) (Figure 2).

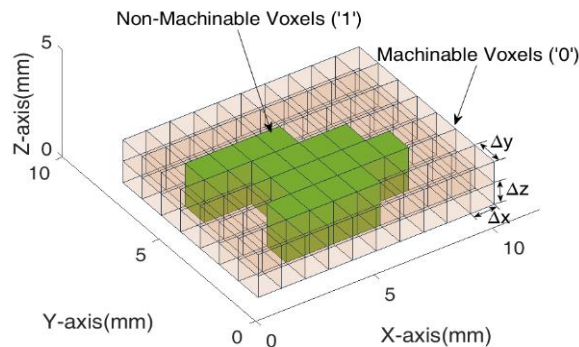


Figure 2: Voxelized part notation.

2.3 Pre-Processing

Pre-processing of the voxel part model is done to preclude the generation of fallacious toolpath. The process is explained below.

2.3.1 Generation of offset region

The offset region is generated on the voxel part model to append the finishing stock allowance and the tool radius compensation. The proposed system does this task by expanding the non-machinable region ($X+/-$, $Y+/-$) in each layer. For instance, Figure 3 (a) shows the initial voxels in the cross-section layer X - Y , every voxel is surrounded by eight neighboring voxels in each layer. To perform the offset by one voxel, all the neighboring machinable voxels '0' of non-machinable voxel '1' are converted to '1' (Figure 3 (b)).

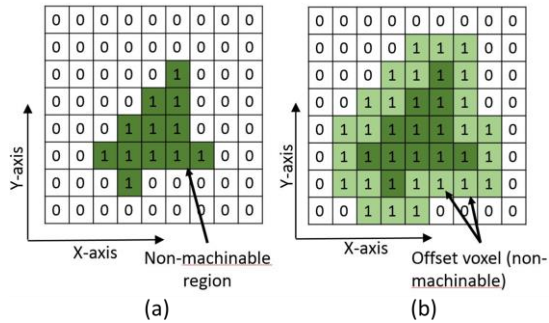


Figure 3: Offsetting of Voxels (a) Initial Voxels, (b) Offset Voxels.

Geometric construction was used to derive the number of voxels to offset, as shown in Figure 4. The cutting tool is placed at the center of the machinable voxel that overcuts the nearby voxels. The amount of material left after the overcutting caused by the cutting tool is the stock allowance.

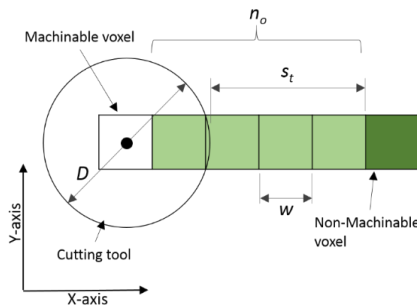


Figure 4: Number of voxels to leave as stock allowance.

The number of voxels to offset is given by Equation (3.1).

$$n_o = \left\lceil \frac{2s_t + D - w}{2w} \right\rceil \tag{3.1}$$

Where n_o represents the number of offset voxels, s_t is the stock allowance, and w is the width of a voxel in the direction of tool movement (here, Δx), and D is the diameter of the cutting tool. For the present work, the voxel size is taken to be equal in both X and Y directions.

2.3.2 Voxel projection

The inner (Blind) features on the part are inaccessible for the vertical (Z) tool in the 3 axis CNC machines (Figure 5 (a)). These inner features are represented by machinable voxels '0' in the present notation. To forbid cutting tool to move over such voxels (prevent overcutting), voxel projection is performed by the union of upper voxel layer to the consecutive lower layer (Figure 5 (b)).

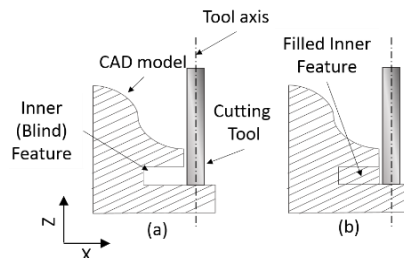


Figure 5: (a) Initial CAD model, (b) Final CAD model after voxel projection.

2.4 Tool Path Generation

The pre-processed voxel data is used to generate toolpath in three steps, viz. Machining area segmentation, Clustering of CL (Cutter Location) points, and Tool path stitching. These are explained as follows.

2.4.1 Machining area segmentation

The cutting tool gets lifted numerous times to machine a part having several machining features. A new way of machining area segmentation is proposed here to reduce the number of tool lifts. For illustration, a closed volume STL part model (500×500×100 mm) with four protruding features is taken as shown in Figure 6(a). The algorithm is as follows.

1. Start moving in the positive X -direction from an initial voxel (part zero for the first region shown by the blue color in Figure 6 (b)) while checking subsequent voxels. If $vx'I'$ (non-machinable voxels) is encountered, find nearest $vx'O'$ (machinable voxels) in the vicinity of the current voxel in the next Y layer and move to that voxel.
2. Keep following the zig-zag path until the traversed voxel is the Y -extremity or a dead-end (i.e., the current voxel is surrounded by $vx'I'$ only). The current position marks the end of the region. Store all the traversed points as the CL points for this region.
3. Find the remaining machining area (Trapped area) and mark the lowermost extremity of the Trapped area as the starting (initial) point of the next region.
4. Repeat steps 1 to 3 until the whole machinable area is covered.

To quantify the decrease in the number of segments, our result (Figure 6 (b)) was compared with the one generated by the area decomposition method reported by Li et al. [18] (Figure 6 (c)). It was seen that for the present case, our approach significantly reduced the number of segments from 13 to 5. The subscripts a and b in Figure 6 (b) specify the start and end of the region number (n) where $n \in [1, 5]$. The tracing starts at part zero (1a), and it continues in a zig-zag manner until the Y -extremity (1b) to mark the first machining region. The tracing again starts at the lowermost point (2a) of the remaining Trapped area, which stops when at the dead-end 2b. The process is repeated to form regions 3, 4, and 5. The voxel positions saved during the tracing form a toolpath. This approach of toolpath generation is termed as Trapped Area (TA) method in this paper. The TA method, however, is seen to induce unnecessary air cutting during machining. As a result, a sequencing strategy is used to visit these regions in an optimized manner to reduce air cutting.

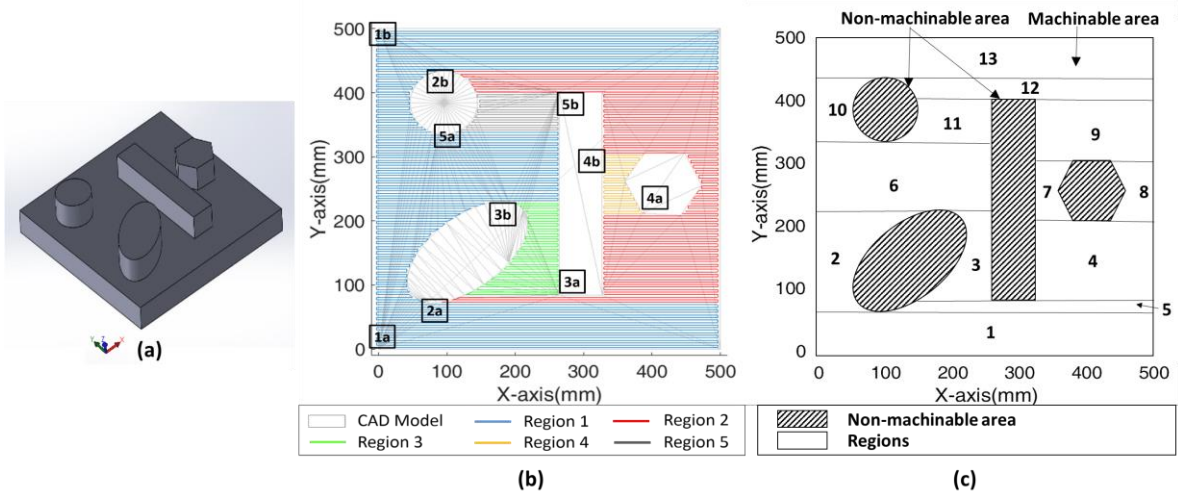


Figure 6: (a) CAD part model, Machining Area Segmentation: (b) Proposed approach, (c) Area Decomposition [18].

2.4.2 Clustering of CL points

The generated CL data is approximately equal to the total number of machinable voxels. This data will create a very verbose CNC program having a large number of program blocks. The clustering of CL points is done to join small voxel-length segments into longer ones and generate a compact part program (Figure 7). The steps of clustering are as follows:

1. Identify the CL points that lie on the same line along the tool feed direction (say X).
2. Select the endpoints of the feed line to eliminate the intermediate points.

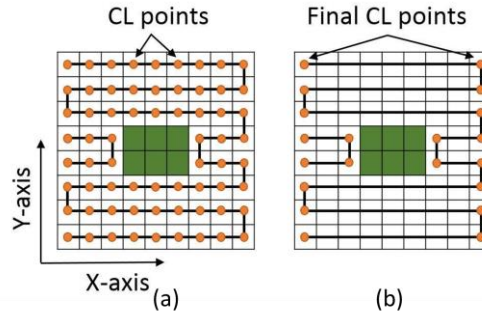


Figure 7: (a) Initial CL data, (b) Final CL data after clustering.

2.4.3 Tool path stitching

After clustering the CL points, the next task is to join the segmented regions in a proper sequence to minimize the air cutting time. The task of finding out the optimum toolpath can be considered as a modified Travelling salesman problem (TSP). In the present case, each region has two points (start and end), and so the total number of routes becomes $n! * 2n$ [8], which makes it harder to solve. Therefore, a modified Heuristic-Genetic Algorithm is used to solve this problem.

The problem can be defined as follows. The region numbers are defined from 1 to n , and with each region, there is an associated binary variable, which is defined as (Equation (3.2)):

$$k_{ij} = \begin{cases} 1 & \text{if the path is connected from region } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

The objective function for the TSP can be stated as (Equation (3.3)):

$$\min \sum_{j=1}^n \sum_{i=1}^n c_{ij} k_{ij} \quad (3.3)$$

Where c_{ij} is the minimum Euclidian distance between regions i and j . The constraints which make sure that a region is traveled only once are given as (Equation (3.4)):

$$k_{ij} = \begin{cases} \sum_{i=1, i \neq j}^n k_{ij} = 1 & \forall j \\ \sum_{j=1, j \neq i}^n k_{ij} = 1 & \forall i \end{cases} \quad (3.4)$$

The genetic algorithm (GA) is used to solve the sequencing problem with two ends per region.

2.5 Implementation of GA

The primary objective of the GA is to provide a solution to the optimization problems with a large solution space in the polynomial time. It works on the probabilistic transition rules rather than

deterministic rules. Therefore, it is better suited to the TSP problem [8]. The steps to solve the tool pat optimization problem in the current work are as follows:

1. Create the initial population of the chromosomes. The elements of the chromosome signify the region numbers, and the order of their placement in the chromosome specifies the sequence of visiting the region. In the present work, the population size is chosen to be between 20 to 100 depending upon the number of regions to be visited.
2. Find the fitness of each chromosome of the population. In this work, a new way of calculating the fitness function to solve the TSP for a “two ends per city” problem is devised. The route indicators ‘a’ and ‘b’ for start and end points of a region respectively are added for every chromosome. After the optimization, the route indicator for each region is also received along with the best route as an output. The fitness f_R of the route (chromosome) R is given by $1/C_R$. Where the C_R is the minimum distance (Airpath length) required to traverse a route R as shown in Figure 8. Equation (3.5) mathematically represents the minimum distance function (C_R).

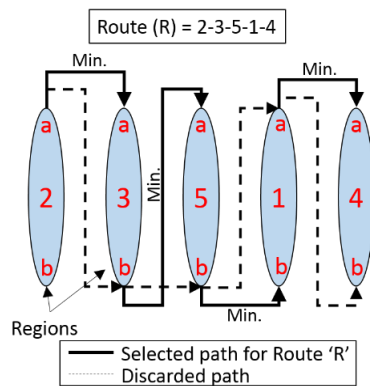


Figure 8: Fitness computation.

$$C_R = \sum_{i=1}^{m-1} \left(\min \sqrt{x_{i+1,k}^2 - x_i^2 + y_{i+1,k}^2 - y_i^2}, k \in [0,1] \right) \tag{3.5}$$

Where m is the length of the chromosome, i is the index for region number in a route, and k specifies the traversing direction of the path for region number $i+1$ (i.e., 0 for start to end and 1 for end to start).

3. Select the chromosomes with higher fitness as parents to reproduce. The rank selection method was used to select parents for crossover [9].
4. Application of crossover: The developed approach uses a partially matched crossover method [2] to preclude revisiting of the same region by preventing duplication of the genes in a chromosome.
5. Mutation: The present approach uses three mutation operators, viz. Swap, Shift, and Flip [14].

The GA was implemented with the maximum number of iterations of 1000, which gives a near-optimal solution for up to 50 region problem. The solution, however, converged rapidly (within 200 iterations) for the problems having less than ten regions.

2.6 Post-Processing

The toolpath generation module generates CL-data, which is processed to generate the final CNC part program in ISO neutral as well as in FANUC (G/M) format.

3 RESULTS AND DISCUSSIONS

The toolpath planning algorithm discussed earlier was implemented in Matlab2015b on a windows 10 platform and run on Intel i5 CPU with 8GB RAM. The performance of the developed system (TA and GA based approach) was evaluated for several case studies with varying part complexities. The results obtained from the GA approach were compared with the toolpath generated using our Trapped Area (TA) method and the parallel zig-zag roughing strategy of the commercial software (NX CAM 12.0.2).

3.1 Case Study 1

The part shown in Figure 6 (a) is taken as a basic test case for the proposed system. The toolpath parameters were chosen to be 10 mm flat end mill, with the sidestep of 6.6 mm, 2 mm depth of cut, and 3 mm stock allowance. The toolpath was generated using both our approaches (TA and GA). The segmented regions (Figure 6 (b)) were joined using GA and the optimum route was found to be *1a-1b-2b-2a-3a-3b-5a-5b-4b-4a*.

Figure 9 (a), (b), and (c) shows the roughing toolpath for a typical layer generated using NX CAM 12.0.2, TA, and GA respectively. Both TA and GA approaches are seen to give a smaller number of lift offs (four) compared to NX (five). The GA approach is better than TA and NX as it optimizes the sequence of visiting the segmented regions to reduce air cutting. This is indicated by the small air path segments in Figure 9 (c).

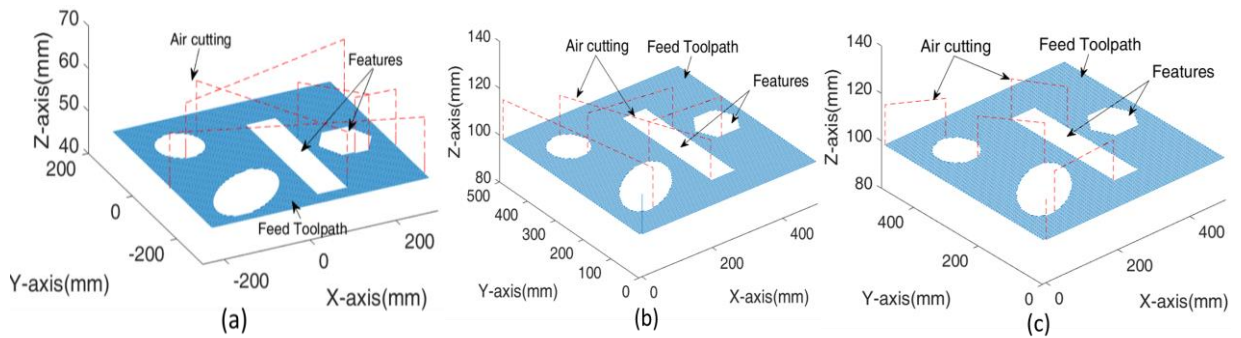


Figure 9: Toolpath for a single layer (a) NX, (b) TA, (c) GA.

Table 1 presents a comparison of the performance of our algorithms (TA and GA) with the commercial CAM software (NX CAM) for machining a single layer.

Machining strategy		Air cutting path length (mm)	Total Path Length (mm)
Our Algorithm	TA	1225	36169
	GA	561	35505
NX CAM		1647	36611

Table 1: Comparison of strategies - Case Study 1.

Both TA and GA algorithms give better results than the NX CAM in terms of the performance parameters like the number of tool lifts, and air cutting path length. Using the GA approach, the air cutting path length is reduced by about 54% than the Trapped area (TA) method and 65% less than the NX CAM.

3.1.1 Effect of part orientation

The size and number of the trapped area changes with the part orientation and so the optimum toolpath would change accordingly. It was decided to study the effect of part orientation on the total cycle time. Figure 10 (a-f) shows the variation in the region segmentation for different orientations of the workpiece for various rotation angles ($\lambda = 20\text{-}90$ degrees with the X-axis). It was seen that the path length changes because the total number of the zig-zag paths get reduced. The maximum reduction in total path length was observed to be 20% at 45 degrees. However, it is difficult to generalize this result because it depends on various factors, such as part geometry and complexity, part size, number and orientation of the features, etc. The proposed approach, however, suggests that the direction of toolpath is a critical factor in finding out the optimum path.

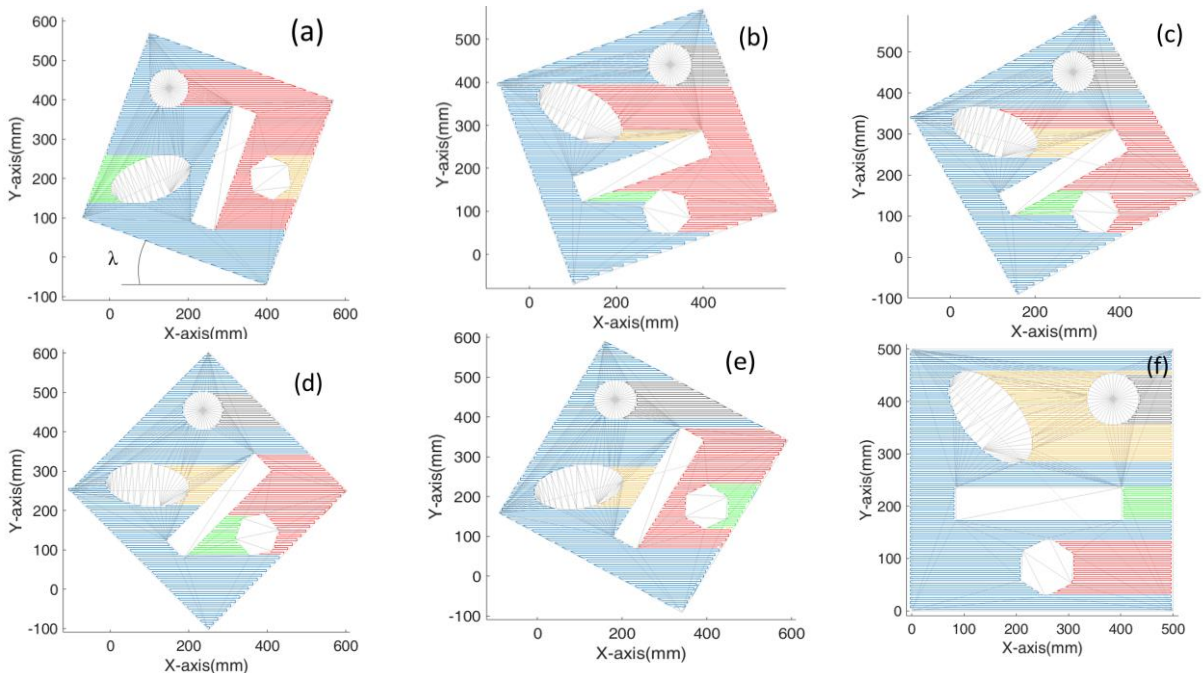


Figure 10: Effect of part orientation (a) 20° (b) 30° (c) 45° (d) 60° (e) 70° (f) 90°.

3.2 Case Study 2

To test the robustness and efficacy of the developed system, two more complex CAD part models were chosen - Part 1 from the NIST repository [22] (Figure 11 (a)), and Part 2, a complex free form surface model (Figure 11 (b)) having several peaks and valleys. The overall size of Part1 is 780×500×200 mm and that of Part 2 is 100×100×100 mm. The toolpath parameters were kept the same for Part 1 as in the previous case study except the depth of cut, which was taken to be 3.96 mm. For Part 2, the sidestep was chosen as 2 mm and depth of cut of 1 mm. Figure 11 (b) and (d) shows the simulated machined component in Vericut using the toolpath generated by our GA based approach.

Table 2 presents the comparison of the performance of our algorithms (TA and GA) with the commercial CAM software (NX CAM). The results show that the GA based approach gives better results in terms of the performance parameters. In particular, the GA based approach has about 37% less air cutting path length than the TA approach, and 47% less than the NX CAM. The reduction in the total tool path length for the GA based approach was observed to be about 9% less than TA and 15% less than NX CAM, respectively.

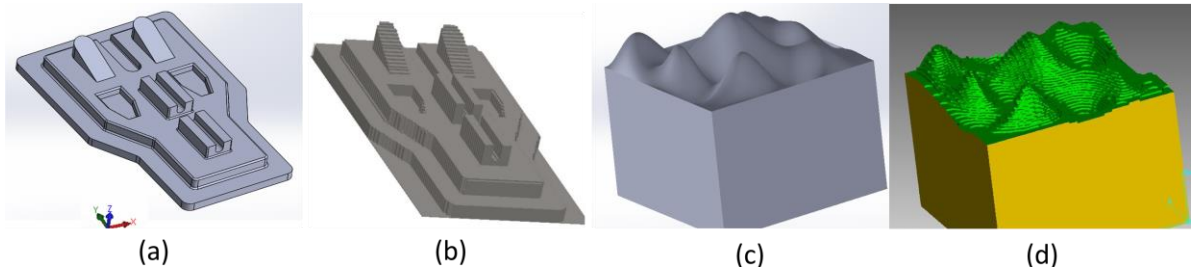


Figure 11: Part 1-(a) CAD model [22], (b) Part 1-Simulated machined component: (c) Part 2-CAD model, (d) Part 2- Simulated machined component.

Machining strategy		No. of Lift offs		Total Air cutting path length (m)		Total Path length (m)	
		Part 1	Part 2	Part 1	Part 2	Part 1	Part 2
Our Algorithm	TA	195	326	90.40	38.20	1072.7	156.2
	GA	195	326	59.94	24.11	999.8	142.0
NX CAM		219	435	96.99	45.52	1079.6	167.4

Table 2: Comparison of strategies for Part 1, and Part 2.

The performance of our algorithm was tested by running the code on Intel i5 CPU with 8GB RAM. It was observed that for the freeform surface part model by decreasing the voxel size from 5mm to 1 mm, the run time increases from 0.40 sec to 15 secs to generate the CNC part program from voxel CAD model. This is thought to be fairly fast and accurate.

The result of the case studies shows that the developed system (GA based) provides better toolpath in terms of the number of tool lifts, air cutting path length, and the total path length. The actual variation in the performance parameters depends upon the part complexity and the number of features.

4 CONCLUSIONS

The current approaches for toolpath generation are often sub-optimal and time-consuming. This paper presents a novel approach to generate a zig-zag toolpath for rough machining operations on a 3 axis CNC milling machine. The proposed methodology is efficient, robust, and computationally inexpensive. The algorithm uses the voxelized part model and segregates the machinable and non-machinable areas. GA based strategy was used to sequence the segmented regions and generate the optimum toolpath with minimal tool lifts. Case studies with various machining features were carried out to compare the performance of toolpath generated by our system with the commercial CAM software. Our system was found to provide more efficient and robust toolpath in terms of a reduced number of tool lifts, air cutting path length, and total path length. The approach can be extended further to other path planning strategies for multi-axis CNC machining.

Aman Kukreja, <http://orcid.org/0000-0001-6060-0890>

Mandeep Dhanda, <http://orcid.org/0000-0003-1168-9299>

S. S. Pande, <http://orcid.org/0000-0001-8805-1854>

REFERENCES

- [1] Abu Qudeiri, J.;Raid, A.;Jamali, M.;Yamamoto, H.: Optimization Hole-Cutting Operations Sequence in CNC Machine Tools Using GA, 2006 International Conference on Service Systems and Service Management, 1, 2006, 501–506. <https://doi.org/10.1109/ICSSSM.2006.320513>

- [2] Al-dulaimi, B. F.; Ali, H. A.: Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA), World Academy of Science, Engineering and Technology, 38(1), 2008, 296–302. <https://www.waset.org/journals/waset/v14/v14-53.pdf>
- [3] Balabokhin, A.; Tarbutton, J.: Automatic generalized cutter selection for finishing of free-form surfaces in 3-axis CNC milling by “surface tolerance and tool performance metrics,” International Journal of Advanced Manufacturing Technology, 93(1–4), 2017, 423–432. <https://doi.org/10.1007/s00170-017-9986-9>
- [4] Balabokhin, A.; Tarbutton, J.: Iso-scallop tool path building algorithm “based on tool performance metric” for generalized cutter and arbitrary milling zones in 3-axis CNC milling of free-form triangular meshed surfaces, Journal of Manufacturing Processes, 28, 2017, 565–572. <https://doi.org/10.1016/j.jmapro.2017.04.025>
- [5] Barclay, J.; Dhokia, V.; Nassehi, A.: Generating Milling Tool Paths for Prismatic Parts Using Genetic Programming, Procedia CIRP, 33, 2015, 490–495. <https://doi.org/10.1016/j.procir.2015.06.060>
- [6] Castelino, K.; D’Souza, R.; Wright, P. K.: Toolpath optimization for minimizing airtime during machining, Journal of Manufacturing Systems, 22(3), 2003, 173–180. [https://doi.org/10.1016/S0278-6125\(03\)90018-5](https://doi.org/10.1016/S0278-6125(03)90018-5)
- [7] Essink, W. P.: CNC Milling Toolpath Generation Using Genetic Algorithms, Ph.D. Thesis, University of Bath, <https://researchportal.bath.ac.uk/files/187928142/WesleyEssinkThesis.pdf>.
- [8] Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Boston, MA, 1989.
- [9] Goldberg, D. E.; Deb, K.: A comparative analysis of selection schemes used in genetic algorithms, In Foundations of genetic algorithms, 1, 1991, 69–93. Elsevier.
- [10] Hajad, M.; Tangwarodomnukun, V.; Jaturanonda, C.; Dumkum, C.: Laser cutting path optimization using simulated annealing with an adaptive large neighborhood search, The International Journal of Advanced Manufacturing Technology, 103(1–4), 2019, 781–792. <https://doi.org/10.1007/s00170-019-03569-6>
- [11] Jang, D.; Kim, K.; Jung, J.: Voxel-Based Virtual Multi-Axis Machining, The International Journal of Advanced Manufacturing Technology, 16(10), 2000, 709–713. <https://doi.org/10.1007/s001700070022>
- [12] Karuppanan, B. R. C.; Saravanan, M.: Optimized sequencing of CNC milling toolpath segments using metaheuristic algorithms, Journal of Mechanical Science and Technology, 33(2), 2019, 791–800. <https://doi.org/10.1007/s12206-019-0134-3>
- [13] Kim, B. H.; Choi, B. K.: Machining efficiency comparison direction-parallel tool path with contour-parallel tool path, Computer-Aided Design, 34(2), 2002, 89–95. [https://doi.org/10.1016/S0010-4485\(00\)00139-1](https://doi.org/10.1016/S0010-4485(00)00139-1)
- [14] Kirk, J.: Traveling Salesman Problem - Genetic Algorithm, MATLAB Central File Exchange, 2020, <https://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm>.
- [15] Konobrytskyi, D.; Hossain, M. M.; Tucker, T. M.; Tarbutton, J. A.; Kurfess, T. R.: 5-Axis tool path planning based on highly parallel discrete volumetric geometry representation: Part I contact point generation, Computer-Aided Design and Applications, 15(1), 2018, 76–89. <https://doi.org/10.1080/16864360.2017.1353730>
- [16] Lai, Y.; Shen, P.; Liao, C.; Luo, T.: Methodology to optimize dead yarn and tufting time for a

- high performance CNC by heuristic and genetic approach, *Robotics and Computer-Integrated Manufacturing*, 56, 2019, 157–177. <https://doi.org/10.1016/j.rcim.2018.09.006>
- [17] Lasemi, A.;Xue, D.;Gu, P.: Recent development in CNC machining of freeform surfaces: A state-of-the-art review, *Computer-Aided Design*, 42(7), 2010, 641–654. <https://doi.org/10.1016/j.cad.2010.04.002>
- [18] Li, H.; Dong, Z.; Vickers, G.W.: Optimal Tool Path Generation for 2 1/2 D Milling of Dies and Molds, In: Olling, G.J.; Choi, B.K.; Jerard, R.B.: (eds) *Machining Impossible Shapes*, IFIP — The International Federation for Information Processing. 18, Springer, Boston, MA, 1999, https://doi.org/10.1007/978-0-387-35392-0_29
- [19] Lynn, R.;Contis, D.;Hossain, M.;Huang, N.;Tucker, T.;Kurfess, T.: Voxel model surface offsetting for computer-aided manufacturing using virtualized high-performance computing, *Journal of Manufacturing Systems*, 43, 2017, 296–304. <https://doi.org/10.1016/j.jmsy.2016.12.005>
- [20] Lynn, R.;Dinar, M.;Huang, N.;Collins, J.;Yu, J.;Greer, C.;Tucker, T.;Kurfess, T.: Direct Digital Subtractive Manufacturing of a Functional Assembly Using Voxel-Based Models, *Journal of Manufacturing Science and Engineering*, 140(2), 2018, 021006. <https://doi.org/10.1115/1.4037631>
- [21] Nassehi, A.;Essink, W.;Barclay, J.: Evolutionary algorithms for generation and optimization of tool paths, *CIRP Annals*, 64(1), 2015, 455–458. <https://doi.org/10.1016/j.cirp.2015.04.125>
- [22] NIST: Fully-Toleranced Test Cases. <https://www.nist.gov/el/systems-integration-division-73400/mbe-pmi-validation-and-conformance-testing-project>
- [23] Oysu, C.;Bingul, Z.: Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining, *Engineering Applications of Artificial Intelligence*, 22(3), 2009, 389–396. <https://doi.org/10.1016/j.engappai.2008.10.005>
- [24] Pavanaskar, S.;Pande, S.;Kwon, Y.;Hu, Z.;Sheffer, A.;McMains, S.: Energy-efficient vector field based toolpaths for CNC pocket machining, *Journal of Manufacturing Processes*, 20, 2015, 314–320. <https://doi.org/10.1016/j.jmapro.2015.06.009>
- [25] Shen, F.;Tarbutton, J.: A Voxel Based Automatic Tool Path Planning Approach Using Scanned Data as the Stock, *Procedia Manufacturing*, 34, 2019, 26–32. <https://doi.org/10.1016/j.promfg.2019.06.110>
- [26] Simonca, F.;Susca, M.;Dobra, P.: Optimization for CNC pathing, 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2018, 1–4. <https://doi.org/10.1109/AQTR.2018.8402761>
- [27] Tarbutton, J. A.;Kurfess, T. R.;Tucker, T. M.: Graphics based path planning for multi-axis machine tools, *Computer-Aided Design and Applications*, 7(6), 2010, 835–845. <https://doi.org/10.3722/cadaps.2010.835-845>
- [28] Tarbutton, J.;Kurfess, T. R.;Tucker, T.;Konobrytskyi, D.: Gouge-free voxel-based machining for parallel processors, *The International Journal of Advanced Manufacturing Technology*, 69(9–12), 2013, 1941–1953. <https://doi.org/10.1007/s00170-013-5148-x>
- [29] Toh, C. K.: Cutter path orientations when high-speed finish milling inclined hardened steel, *International Journal of Advanced Manufacturing Technology*, 27(5–6), 2006, 473–480. <https://doi.org/10.1007/s00170-004-2206-4>