





Clustering Of Machinable Volumes For Tool Selection In 3-axis Milling

Hariharan Krishnamurthy¹ , Manoj W. Bhonge² , Tathagata Chakraborty³ 

¹HCL Technologies Ltd, India., hariharan_k@hcl.com

²HCL Technologies Ltd, India., Manoj.Bhonge@hcl.com

³HCL Technologies Ltd, India., tathagata.chakr@hcl.com

Corresponding author: Hariharan Krishnamurthy, hariharan_k@hcl.com

Abstract. The selection of tools for machining operations is an important part of the process planning task for machining. Many methods have been developed to obtain optimal tool sequences from a mathematical optimization perspective; yet tool selection is largely done by a skilled and experienced machinist today. This paper presents a computational method to aid tool selection for 3 axis milling operations. Given 3D meshes of the input part and available tools, the potential volume *machinable* by a tool is computed on a tight bounding-box as stock. The input shapes are voxelized and the tool placed above the part surface to identify gouge-free voxels computationally, and thus the machinable and leftover volume with the tool. A method is presented to visualize the leftover volume by constructing its bounding surface. A tool is represented using the machinable volume, the Material Removal Rate (MRR) and machining time estimates as criteria or 'features' for clustering. Three methods, namely k-Means, mean-shift and agglomerative clustering, are explored to group the set of tools into clusters, which can be used to compose an optimal tool order. Experiments are performed with example parts considering flatend mill tools. The results indicate that a combination of criteria, e.g. the pair of machinable volume and time estimate, or all three features yield clusters of tools agreeing with intuition for machining a part. A single criteria, such as machining time alone, may not be distinctive and realistic enough. A simplified theoretical model of the variation of machinable volume with tool size is presented. On the whole, the method is aimed at simplifying tool selection by logically grouping tools according to the input shapes and factors pertinent to machining. The method is generalizable to any tool shape and operation. Future work includes consideration of realistic stock shapes, sequence of machinable volumes with operation sequences, extension to tool orientation, and development of the theoretical model.

Keywords: Tool selection, Triangulated meshes, 3-axis milling, Machinable volume, Clustering methods, voxelization, theoretical model

DOI: <https://doi.org/10.14733/cadaps.2023.306-321>

1 INTRODUCTION

Three-axis machining is a commonly used method of part production. The three axes refer to the three simultaneous directions of tool motion, commonly forming a Cartesian co-ordinate system. The machining task starts with the CAD model of the target part by deciding the type and sequence of operations and the tools to be used. Then the appropriate toolpaths are computed which are post-processed into NC code that is executed on machines. An important part of the planning is to identify the set of operations and tools. The selection of tools involves geometry-based factors, such as tool size, the stock, tool materials and the machines' capabilities among many other factors. A typical machining centre would have a multitude of tools of different shapes, sizes and frequency of use, of which only some are suitable for a given job.

Over the years, tool selection has been studied as an optimization problem of finding an optimal subset from a set of tools satisfying objectives such as minimum machining time, energy consumption, maximizing tool life and so on. Minimizing the machining time is most important and desirable in machining practice. Of late, all of the objectives are becoming important at various stages of machining. In spite of considerable research on tool selection and sequencing from an optimization perspective, tool selection is still performed by the machinist based on experience and knowledge. This could be due to the proposed methods being too sophisticated to implement in commercial CAM software, or the types of tools available being specific to machine shops. This paper presents a computational method of tool selection that aids the selection of a set of tools appropriate for producing a given part from triangulated meshes of the set of tools and target part. The volume potentially machinable by a tool is computed by discretizing the part and tool into voxelized models and placing the tool above the part surface to compute interference-free voxels. The computed volume is combined with other criteria important for machining such as the Material Removal Rate (MRR) to represent a tool as data for clustering; a few clustering methods are used to arrive at the clusters or subsets of tools. Thus the choice for tool selection is narrowed into sub-groups in a divide-and-conquer strategy; an optimal subset can be formed by selecting one tool from each cluster. Other considerations and criteria can be incorporated into the clustering process, making the method practical for machining.

The rest of the paper is organized as follows. In section 2, prior art on machined volume computation and tool selection are reviewed. Then in section 3 the voxelization of the input shapes and computation of removable volume are described. Section 4 discusses the 'features' to represent tools and the clustering methods for tools. The results of clustering are deliberated in section 5. A theoretical model of machinable volumes is briefly developed in section 6. The paper is concluded in section 7, with future work directions.

2 LITERATURE REVIEW

The selection of optimal tools for machining has been studied extensively. The problem has been researched mainly as a part of process-planning, where the sequence of operations and tools for machining a part are devised; the research mainly focussed on subtractive manufacturing. Initially, researchers considered 2.5 axis milling operations, and later 3-axis milling of constant-depth pocket regions. The work presented in [15] is a method for obtaining flatend mill tool sequences with least machining time for planar pockets through a branch-and-bound exploration of the tree of sequences.

In [23], the concept of a 'Voronoi mountain', the extrusion of points in a planar pocket proportional to their distance from edges of the polygon's Voronoi diagram, is proposed. Using a Voronoi mountain, the toolpath and consequently the leftover shape for the next operation are computed. The total machining time is minimized by evaluating and incrementally adding pairs of tools in a dynamic programming approach. The Voronoi mountain approach is extended in [20] to pockets with many islands and open profiles. Tool sequences represented as paths in a graph with edges weighted by process, tool parameters and a shortest-path algorithm is then used to find the optimal sequence. The work in [4] adaptively slices part meshes to get the shape of blocks of material, which are evaluated for access by tools modelled as cylinders. Combinations of tools from groups based on tool size are evaluated to obtain the sequence with the least machining time and high MRR.

The authors of [11] a planar pocket split into two regions and the representation of all tool sequences in a graph to evaluate sequences based on accessible area and toolpath computation. Subsequently, in [12], the sequence of operations with tools for a prismatic pocket is represented by a graph of intermediate leftover shapes, and tools are evaluated starting from the largest. Heuristics for reducing computation, such as considering sequences ordered by tool size are adopted. Subdivision is also used in [19], where a linear polygon denoting a pocket boundary is decomposed into convex ones, and the toolpath is computed for each polygon using offset and zig-zag segments. The total machining time, including time for link moves and others, is minimized from tabulated machining sequences using dynamic programming. In [9], the feed rates from a machining handbook are used to compute and the optimal tool sequence with four meta-heuristic methods and compare the sequences. The work in [6] aims at choosing optimal cutting parameters using simulation-based evaluation of remaining tool life.

The optimal machining of multiple parts simultaneously is considered in [26]. The optimal sequence is computed from a graph of sequences of the jointly-represented machined regions on all parts. Ant-colony optimization is used with a fuzzy objective function involving tool selection for a flexible manufacturing system in [8]. Roughing of a pocket is treated analytically in [7] to select three tools yielding the least total time from a set of tools. The energy consumption and usage cost of tools in milling operations is described in [25], using tool travel length from accessibility-based toolpath computation. In [13], energy consumption of tools is minimized through mixed-integer programming for obtaining tool sequences in a flexible manufacturing shop.

In the last decade, voxel-based methods have been explored for toolpath computation for 3 and 5-axis milling operations. In [3], greedy optimization of the volume removed, computed through software, and other parameters are used to optimize the average MRR. Planning for producing functioning assemblies is done in [16] through evaluation of feasible tool orientations by placing the tool on offsets of the target parts. The work also presents a voxel-based toolpath generation method. The 3D scan of a stock is used for realistic toolpath computation in [21], by computing the volumetric difference between the stock and the target part and taking planar sections of the difference. Voxel-based methods have been used for simulating machining processes, such as in [17] where the machining forces are predicted using a voxelized representation of the stock.

It is seen from the literature reviewed above that the majority of methods for tool sequence optimization consider 2.5 axis milling of planar features, and represent the problem as finding shortest paths in weighted graphs. For this approach, almost all the methods require the toolpaths themselves. Recently, voxel-based methods have been used for toolpath computation in 3-axis milling; but their main utilization seems to be in the realistic simulation of the machining processes. Unsupervised machine learning techniques such as clustering have been applied to the domain of manufacturing primarily for prognosis and monitoring of processes, such as predicting tool wear. In this paper, we propose to combine the concept of voxel-based computation of the volume machinable by tools with other criteria to group a set of tools into subsets which are easier to analyze. Thus the search space for tool selection is reduced significantly. The method does not require toolpath evaluation, and can be incorporated at the process-planning stages itself so as to quickly aid the machinist in tool selection.

3 COMPUTATION OF MACHINABLE VOLUME

A tool can machine certain regions of a stock to obtain a shape closer to the target part; the extent of the material removable by a given tool depends on the tool's size, its shape and the shape of the machining features on the part. By machining features, we mean the volumes in the stock bounded by the target part's surface, such as pockets, holes, slots and so on. Machining tools come in a large variety of shapes and sizes. Different types are suitable for different operations. Flatend, ballend and hognose tools are commonly used in milling (chapter 32, [22]). Flatend tools are predominantly used for roughing, finishing (when requiring sharp corners) of parts with "extruded" 2.5D features such as pockets and bosses. Ballend or ballnose tools are used with freeform surfaces in 3 to 5-axis machining. Drill tools are meant for hole-machining, entry-drill

operations preceding roughing. Conical end-mills, form cutters like T-slot cutters, barrel tools are examples of specialized tools. Operations and tool types are described in many sources, such as [22]).

The actual material removable by a given tool depends on the tool path apart from part shape and machining parameters, as it specifies all the positions to be taken when machining. However, the volume potentially machined by a given tool can be computed without the toolpath, considering that toolpaths with sufficient passes can be generated to cover all the points reachable by the tool. The potentially removable volume from a given stock by a tool is termed as the machinable volume hence. While the machinable volume is difficult to compute analytically in a general case, a discretized computation is certainly possible. The discretization of the part and tool shapes and the computation of machinable volume is described in the following.

The tool size alone is considered for computing the machinable volume; also only flatend tools are used. The stock for the part is taken to be the tight bounding cuboid of the part shape. A flatend tool is modelled as a cylinder of the diameter being the tool size. An example of an input part and tool shape are shown in Figures 1(left) and 1(right) respectively. Hence, the tool size refers to the diameter of the tool.

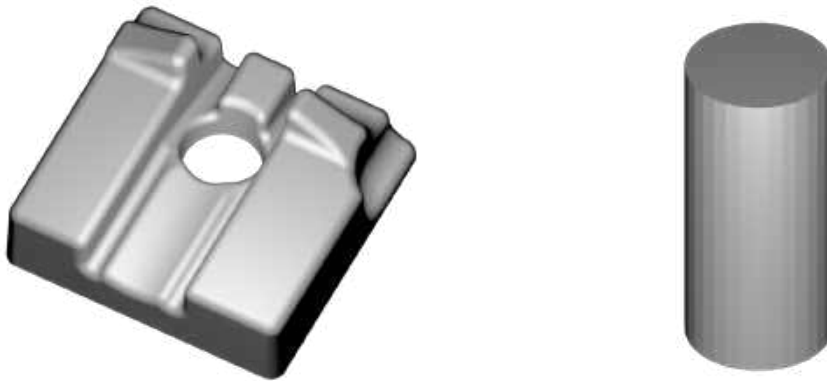


Figure 1: Examples of input shapes: (left) the target part, and (right) a flatend tool of 10 mm diameter.

3.1 Voxelization For Volume Computation

Voxelization is the process of constructing a discretized representation of a shape. Voxelization results in representing the shape by a set of voxels or volume elements. For this, the input part shape is enclosed by a uniform grid. The cells of the grid are classified to be one of three states, namely contained by the part, completely outside the part and intercepting the part; the last state means that the part shape intersects a cell. The grid cells intercepting the triangles of the part shape are termed as boundary cells. It is noted that grid cells denote voxels when containing material (part or the stock). The identification of boundary cells is performed through the ‘triangle-box’ overlap test in [2].

Given an input shape, boundary cells are identified first. Then the other cells of the grid can be classified as being inside or outside the part shape. The range of cells occurring between two boundary cells will be within the part, such as two opposing surfaces. The cells not contained in this range will be outside the part. In this paper, it is assumed that one pair of boundary cells would occur at a given (x, y) location of the grid. In other words, disjoint ranges of cells are not considered, meaning no ‘undercut’ regions with regions overhanging from below are expected. This is a reasonable assumption for 3-axis milling, as common tools cannot reach undercut regions due to the risk of collision from non-cutting portions like the holder. The classification of grid cells is done for the tool and part shapes individually. The grid enclosing a shape (part or tool) is constructed

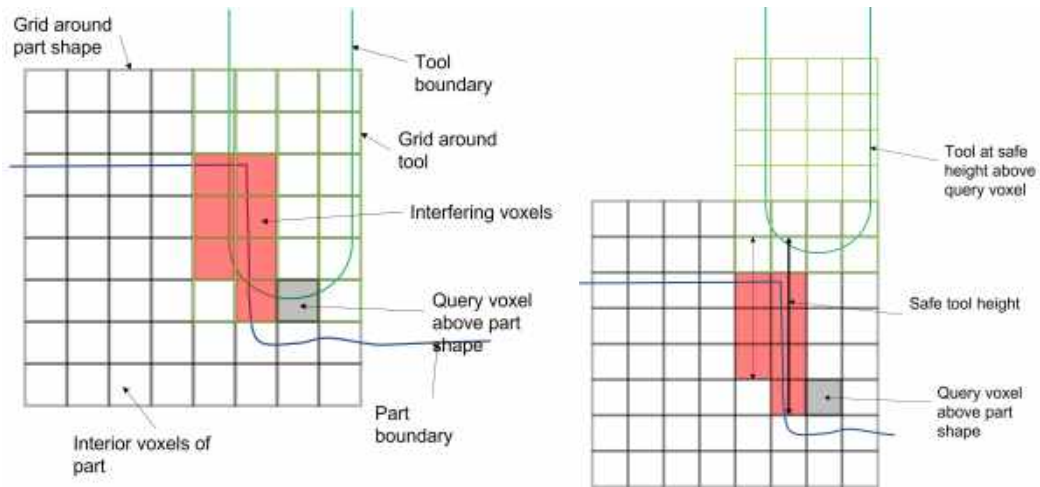


Figure 2: Schematic depiction of the interference detection and calculation of safe tool height: (left) interfering tool and part voxels (shaded red), and (right) lifting up of tool to a safe height.

with the dimensions of the bounding box of the shape.

3.1.1 Computation Of Machinable Voxels

The grid with the cells classified as above, is used to compute the accessibility of the part shape by the tool. Let us assume that the voxel containing the centre of the tool's base represents the tool tip. If the tool does not interfere with the surrounding part shape when placed above the part, i.e., the voxel above a part boundary voxel, then that voxel is considered accessible.

If the tool does interfere with the part shape, then the height by which the tool is to be lifted to escape or avoid the interference can be calculated; this height is the 'safe' tool height at that location. The safe height is calculated as follows. When the tool is placed above a part voxel, each column of voxels of the part grid is checked for overlap with the *corresponding* column of voxels of the tool grid. The number of overlapping voxels represents the 'depth' of penetration of the tool into the part in that column. The maximum depth of penetration found out of all columns of the tool grid would be the minimum height to lift the tool to avoid penetration. Essentially, the "Hausdorff" distance between the part and tool boundary voxels, when the tool is placed just over the part surface, is the safe height to lift the tool. The interference of part, tool voxels and the calculation of safe tool height are depicted schematically in Figures 2(left) and 2(right) respectively.

Thus, for each voxel just above a boundary voxel of the part, an interference-free or safe tool height can be computed; if this height is 0 then the voxel is accessible by the tool. When the tool grid is at the safe height in a column, the *exterior* voxels of the part grid *corresponding* to boundary and interior voxels of the tool are considered machinable. The exterior voxels would be the stock material. The union of all such machinable voxels represents the machinable volume. The set of all voxels not reachable by the tool i.e. not covered in the above set represents the set of unmachinable voxels; together they denote volume left over after machining by the tool. So the final set of machinable voxels S_{mach} is $S_{mach} = \cup(S_{acc})$, where S_{acc} is the set of accessible voxels at one location. Hence, the set of leftover voxels is $S_{leftover} = S_{ext} - S_{mach}$, where S_{ext} is the set of exterior voxels of the part grid.

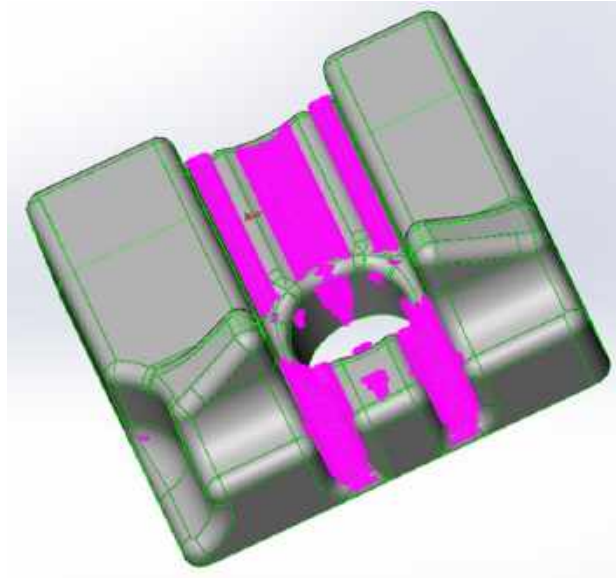


Figure 3: Visualization of leftover volume on a part on machining with a tool - the leftover (translucent pink) bounding surface with a 16 mm flatend tool shown for the part in Figure 1.

3.1.2 Voxel Size: A Note

Voxelization is mainly controlled by the resolution, or equivalently the voxel size. Since part dimensions fix the bounding box and XYZ resolution, voxel size is chosen beforehand. A large voxel size leads to a coarse representation and the machinable volume, while a small size gives higher accuracy; however it also incurs higher computational cost. A sub-millimeter accuracy is desirable at the least for modelling and machining simulation, so a voxel size of 0.8 mm is used, to balance accuracy and computational cost. Experiments with different voxel sizes for a part showed the same *trend* in machinable volume, so the chosen size is deemed sufficient for demonstration. For higher accuracy, the voxel size can be reduced further with parallelization and GPU-based implementations.

3.2 Visualization Of Leftover Volume

The leftover volume computed with a given tool can be visualized on the part shape as a surface; this is the leftover expected on machining away all the machinable volume. The bounding surface of the leftover volume needs to be constructed to visualize the leftover volume. For this, it is observed that interior voxels of a type (leftover or machinable) would be completely surrounded by voxels of the same type. The voxels on the boundary would only partially surrounded by voxels of a type. Thus, the fraction of *leftover* voxels surrounding a given leftover voxel is used to assign a value to that voxel. The value ranges from 1.0 for interior leftover voxels to ~ 0.0 for voxels on the boundary; so, the value is a scalar varying in space. The iso-surface of the scalar for a small value can denote the bounding surface; presently the 'small' value is taken as 0.01. Since all volume computations are done 'in-situ' on the part shape, the iso-surface is visualized on the part. For e.g., Figure 3 shows the leftover on a CAD model. It is noted that the computations are integrated into a CAM add-in for a CAD system; so the leftover can be visualized in the CAD system conveniently.

4 CLUSTERING OF MACHINABLE VOLUMES

Clustering is broadly defined as the grouping of elements of a given set into distinct subsets [1], based on the similarity and nature of the elements. It is desirable to obtain mutually exclusive subsets; this depends on the distinguishing ability of the features, i.e. attributes, used to represent elements, and to some extent the method of clustering. A number of clustering methods have been researched in the fields of data analysis and machine learning, spurred by the ever-increasing interest in unsupervised machine learning. Of these, we restrict our attention to three methods, namely the K-means method, the mean shift method and the agglomerative clustering methods.

4.1 Features For Clustering

The machinable volume with a tool is used as a feature for clustering the set of tools, as mentioned in the introduction (section 1). A large tool would machine a smaller volume on the stock in general, due to smaller machining features and regions like corners. Such features are reachable by a smaller tool leading to a larger volume being removable by smaller tools.

The rate at which material is removed by the tool during machining is known as the Material Removal Rate (MRR). The MRR depends on the speed of the tool along its trajectory as well as the extent of engagement of the tool's flutes (cutting surfaces) with material. The rate of tool motion while cutting material, referred as the feed rate, depends on the tool size among other factors. Typically, a tool manufacturer recommends the range of feed rates supported by a tool for machining appropriate stock materials. The feed rate is not specified as linear speed, but through two quantities - one the speed at the flute periphery and the other the feed per tooth. The former is the speed at the circumference of the flute and is expressed as Surface Metres per Minute (SMM) or Surface Feet per Minute (SFM). The feed per tooth is the distance moved by the tool during engagement of one tooth of the flute. The linear speed of the tool can be obtained from the speed and feed as shown in equations 1,2 and 3 respectively.

$$S_{rpm} = \frac{SMM * 1000}{\pi * D} \quad (1)$$

$$F = \text{Feed per tooth} * N \quad (2)$$

$$S_{tool} = S_{rpm} * F \quad (3)$$

where:

S_{rpm}, S_{tool} – spindle rotation and linear speed in rev., mm per min. respectively

F, N, D – Feed per revolution, and number of teeth, tool diameter respectively

The feed rate i.e., the linear tool speed, varies along the toolpath during actual machining due to various factors such as shape characteristics of the toolpath, the forces acting on the tool and so on; it is desirable to achieve a constant feed rate throughout the cutting in machining practice. Presently, the linear speed is considered constant throughout the toolpath. The MRR is expressed using the linear speed and the area of engagement. The latter is approximated as the product of the axial Depth-Of-Cut (DOC) and the stepover, the distance between two passes of the tool along the material; stepover is also referred as the radial DOC. These approximations are standard in machining practice (see for e.g. [10]). For the computations to be realistic, a stepover of 50% of the tool size is considered and axial DOC is taken to linearly vary within a range dependent on the range of tool sizes. The computation of MRR is shown in equation 4.

$$MRR = S_{tool} * DOC_{radial} * DOC_{axial} \quad (4)$$

where:

MRR, S_{tool} – Material Removal Rate, linear tool speed respectively
 $DOC_{radial}, DOC_{axial}$ – Radial and axial Depth-Of-Cut respectively

Thus far, we have the machinable volume and the MRR of the tool, albeit approximately. The quotient of the machinable volume with the MRR can provide an estimate of the time for machining, if the entire part were machined from the stock with a tool. The machining time is an equally important feature characterizing the use of a tool as the machinable volume. A large tool, though giving a lesser machinable volume, can support a higher feed rates than a smaller tool that can machine a larger volume. So, even if the smaller tool can machine the entire part by itself, the time required for machining makes it impractical to use the tool alone. The machining time complements the machinable volume as a feature. It is noted that the MRR is independent of the volume being machined, while the time depends on both the MRR and the machinable volume. The machining time is T_{mach} is shown in equation 5, where the machinable volume is V_{mach} ,

$$T_{mach} = V_{mach}/MRR \quad (5)$$

4.2 Method Of Clustering

The clustering exercise can be explored in the following three aspects:

1. Method of clustering
2. Number of clusters expected
3. Dimensionality of the feature space i.e. the number of features representing a tool

As mentioned before, three methods of clustering are considered here. The number of expected clusters depends on the method using the number of clusters as a 'hyperparameter'. There are three features available currently, the machinable volume, MRR and machining time. Combinations of the three features can be used for clustering as follows. Each attribute can be considered solely as singleton combinations. Three *pairs* of attributes are possible, of which the pair of machining time, machinable volume, and machinable volume, MRR is considered; the pair MRR, machining time is redundant to consider. Finally, all the three attributes can be considered representing the tool as a three dimensional vector.

The implementation from the python package scikit-learn ([18]) was used for all clustering methods. The plots were generated with the package matplotlib ([14]), while dendrograms were made with the package scipy ([24]).

4.2.1 Example Parts

Three parts are considered as examples to apply the method presented. The parts are referred to as A, B and C for convenience hence. The part A is the realistic mould/die shown earlier in Figure 1(left). Part B is a synthetic part with nested pockets as shown in Figure 4(left) while part C, shown in 4(right), has multiple 2.5D features like pockets and holes.

4.3 k-Means Clustering

The k-Means clustering method is a well-known, standard method of clustering. The method accepts a set of observations and the hyperparameter k - the number of clusters. It iteratively calculates the centroid, the mean of the tuples representing the observations, of each of the k clusters. There are techniques to help find

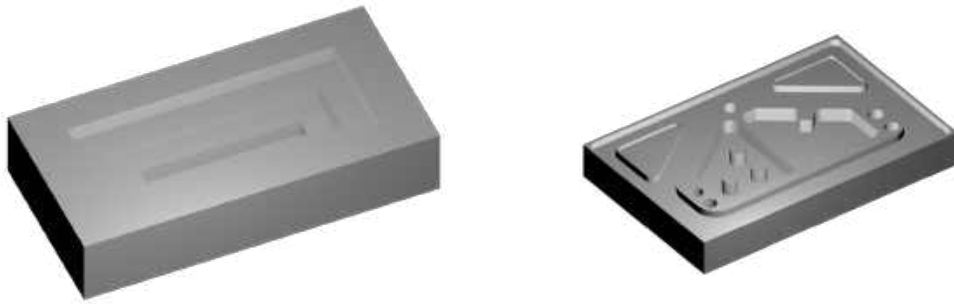


Figure 4: Parts considered for experiments - (left), part B and (right) part C. Part B has stepped pockets while part C has many 2.5 D features. Views of parts chosen so as to show features better.

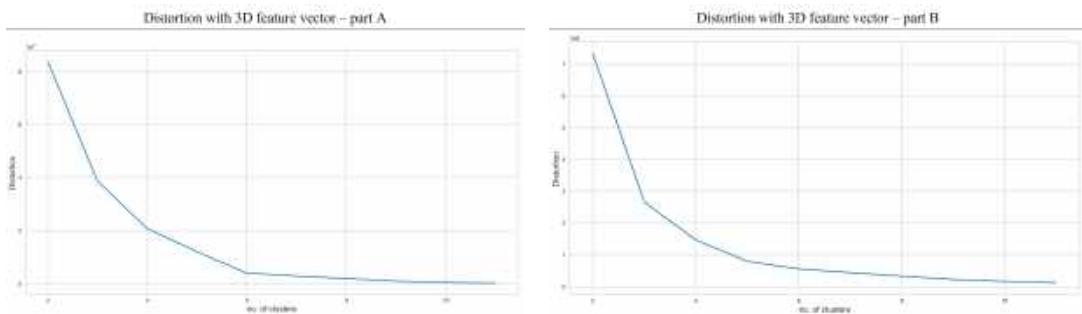


Figure 5: Example of distortion score plots for k-Means for two parts - (left) for part A and (right) for part B. Both exhibit an ‘elbow’ at $k = 3$ or $k = 4$.

the ‘optimal’ k such as the ‘elbow’ method based on a distortion score (see for e.g. [5]) or a silhouette score and so on. Since these are purely data-based techniques, the context or particular application would ultimately help arrive at the appropriate number of clusters.

Currently, the silhouette score is plotted for a range of ‘ k ’ values for each feature-representation of a tool, from singleton to all three dimensions. The same exercise is performed for the distortion score. The silhouette score is consistently reported as being maximum for $k = 2$, but expecting two clusters did not justify an elaborate clustering exercise for the example parts. The distortion score gives a more or less clear ‘elbow’ at $k = 4$ and $k = 5$ clusters in most of the cases; for e.g. Figure 5(left), has an elbow at 4. So, k is chosen as 5 for part A.

The synthetic nature of part B, having three pockets, implies that the machinable volume would be distributed at three levels, suggesting k to be 3. The calculation of distortion score, shown in Figure 5(right), suggests elbows at $k = 3$ and $k = 4$ in some cases. The value of 4 is chosen to conservatively use the distortion score and knowledge of the part shape. The plots of distortion score for part C also exhibit elbows at $k = 3$ and $k = 4$. The value of 4 is chosen again, in a lenient manner.

The number of clusters is not a hyperparameter in mean-shift clustering. A variable called the ‘quantile’ is a hyperparameter. Presently, the default quantile in the scikit implementation is used.

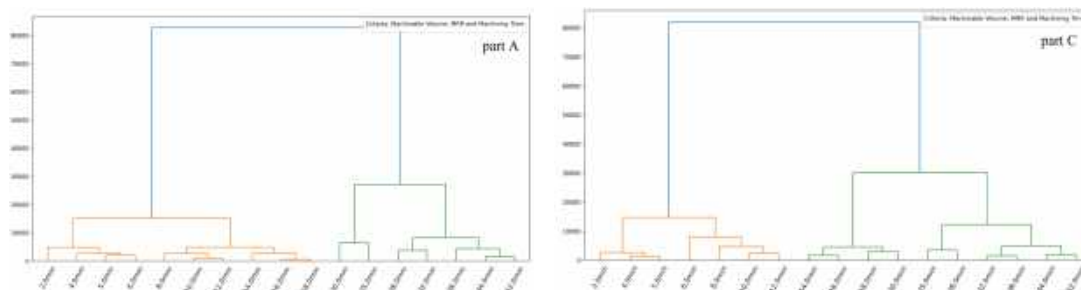


Figure 6: Dendrograms with 3-D feature vector for two parts - (left) for part A and (right) for part C. In both cases, branching increases after 4-5 branches from the top.

4.4 Agglomerative Clustering

Agglomerative clustering is a general and long-known clustering method. Here, clusters are formed from individual data points by successively merging neighbouring points into clusters, and clusters into bigger clusters until all data points are covered. Neighbourhood is decided by a measure of proximity/distance. The method of connecting clusters based on a quantity or criteria is called a linkage method. Different linkage methods and distance metrics lead to variants of agglomerative clustering. Currently, the 'ward' linkage method with Euclidean distance between points is used. A graph representation of the linkage method for the set of data points, starting from clusters of single data points as leaf nodes, is called a dendrogram. A dendrogram can guide the choice of number of clusters. The length of edges in a dendrogram reflects the "distance" between the clusters.

As before, the dendrogram is constructed for each combination of features, a sample of which is shown in Figure 6(left), for part A. From 6(left), it is seen that the distance between the branching points reduces after four to six branches. So, the number of clusters is chosen to be 5 to consider all the dendrograms. For parts B and C, the dendrograms show a greater spread at 4 or 5 clusters, so 4 is chosen, also to consider knowledge of the part shape; the example for part C is shown in figure 6(right).

5 RESULTS AND DISCUSSION

The results of clustering of a set of tools with each of the three example parts (section 4.2.1) are presented here. The tools considered are flatend mill tools with size ranging from 2 mm to 52 mm, a set of 17 tools is used. In all the figures, the set of tools is plotted as points with the attributes used as dimensions, against the tool size, before and after clustering. The tools belonging to a cluster are depicted by identical colour. The parameters of voxelization are as discussed in sections 3.1, 3.1.2. The tool size and machinable volume are in mm, mm^3 units respectively, while the MRR is expressed in mm^3/min . The machining time is given in minutes.

The results of clustering for part A under the dimension of machinable volume, MRR and machining time are shown in Figure 7, with the k-Means method. For brevity, the clustering for the combination of all three features alone is shown. It was found that the clusters under machinable volume, the pair machinable volume, machining time, and all three features roughly agree. From the tools <10 mm fall into one cluster, between 10 and 25 mm into two clusters and above 25 mm into two clusters. The results of agglomerative clustering are very similar to those obtained with k-Means. The mean shift method shows a slightly differing trend of three clusters, generally, with tools <20 mm in size in one cluster, >20 & <30 mm in one and the rest into another cluster, as shown in Figure 7(below).

The results of clustering for part B are shown in 8, again only for the vector of all three features. The

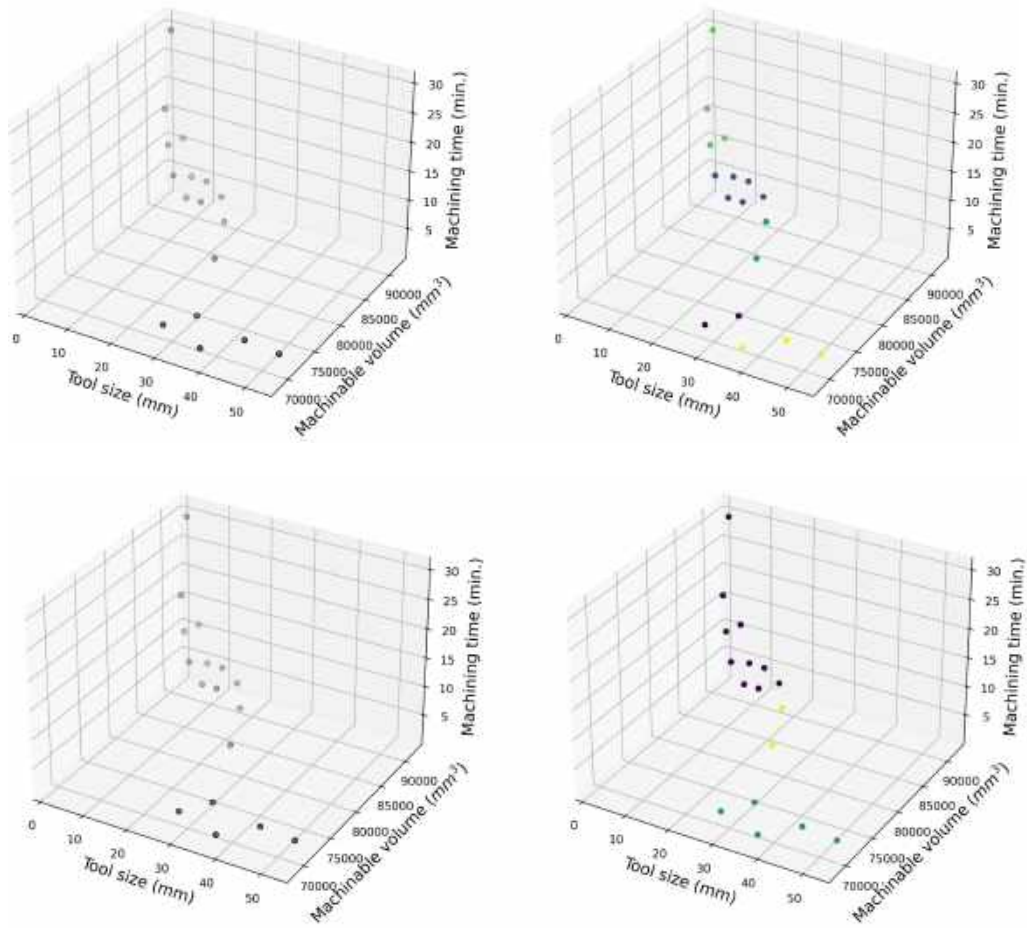


Figure 7: Results of clustering for part A - (above) for the k-Means and (below) for the Mean Shift method. (Left) input points, and (right) points coloured by cluster. Agglomerative clustering gives similar results to k-Means.

clustering is unanimous for this part with tools falling in three distinct groups, $0 < D < 10$ mm, $10 < D < 20$ mm and > 20 mm. There are differences observed in the subdivision of one of the clusters into two, the differences being in the range of tool sizes being split. As with part A, the results of k-Means and agglomerative clustering agree to a large extent. The mean shift method yields three clusters with the same grouping as the other methods, as shown in Figure 8(below).

The clusters obtained with part C are shown in Figure 9. The four clusters obtained are also rather uniform across the methods, with tools < 8 mm, $8 < D < 18$ mm forming two major clusters, one to a few tools in the range of $20 < D < 28$ mm, and all the rest in the last cluster. The results of agglomerative and k-means clustering match. Not surprisingly, mean shift also yields similar results; this leads to all methods agreeing on the clusters, unlike the other two parts.

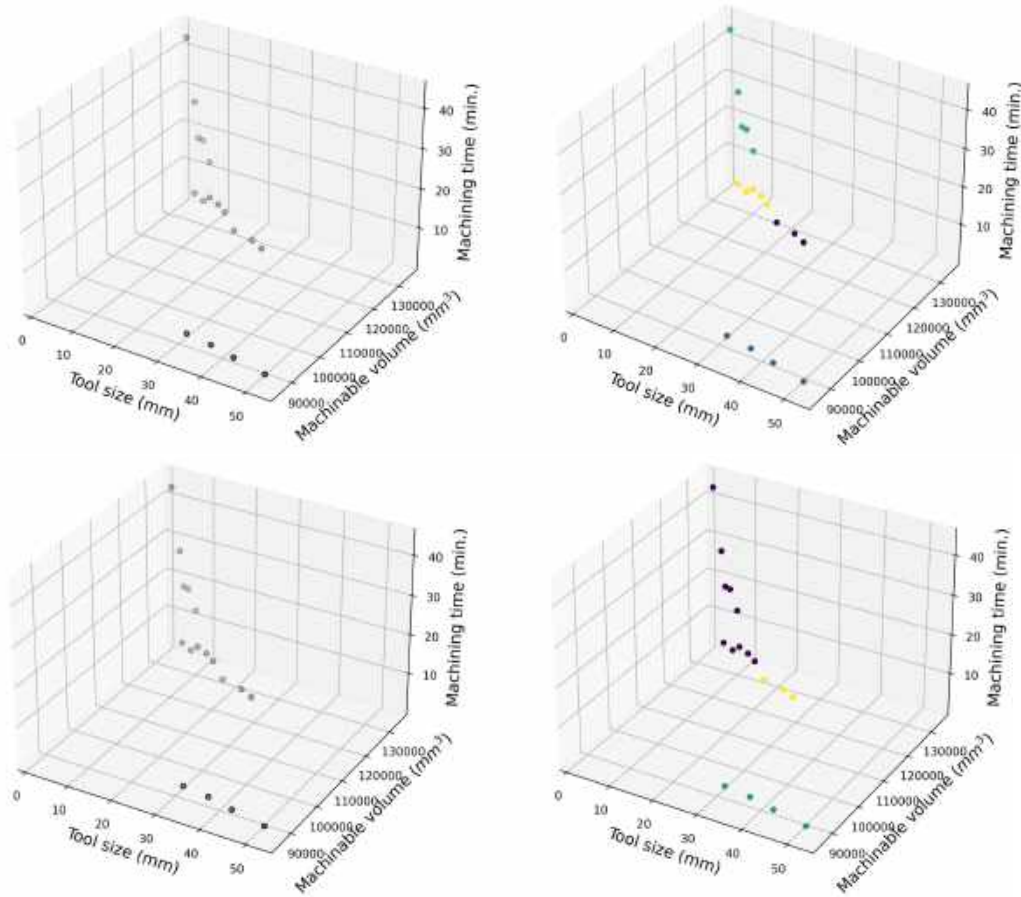


Figure 8: Results of clustering for part B - (above) for the k-Means and (below) for the Mean Shift method. (Left) input points, and (right) points coloured by cluster. The distinct clusters corresponding to the pockets are seen.

5.1 Discussion

The clusters obtained with part A roughly agree with the expected grouping of tools. The clusters with the singleton machining time differ significantly from the clusters with other combinations of features. This could be because of the difference in machining time between two tools being not distinctive as the corresponding differences of other features.

The synthetic nature of stepped pockets on part B implies that the clustering results are completely predictable. Here too, the machining time as a singleton feature gives differing results from the other combinations. The marked distinction of clusters is seen in the results, where the tools are distributed in three 'steps' or distinct levels.

A major observation on the results for part C is that tools sized > 20 mm cannot machine any volume on the part. That is the machinable volume is 0 for these tools, so the machining time is also 0. In effect, there are three major clusters in the range $0 < D < 20$ mm, and the fourth cluster containing one feasible tool at most. The useful aspect of clustering is the distinction shown between the tools with a non-zero machinable

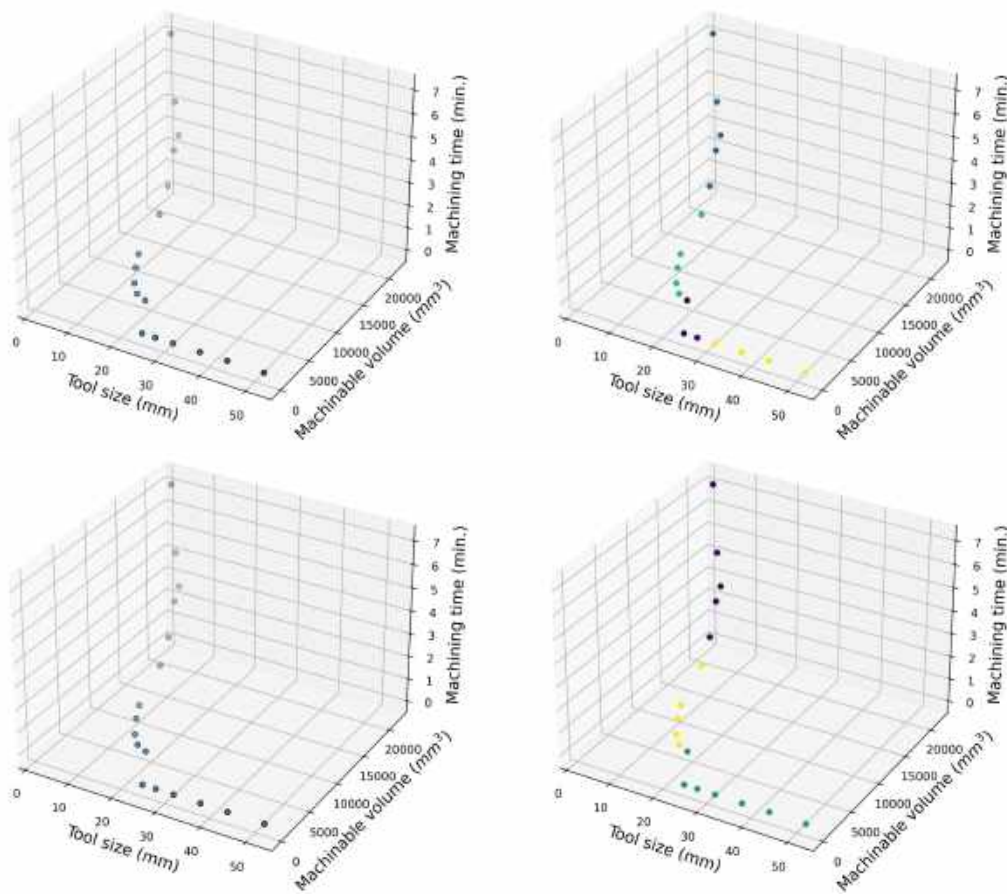


Figure 9: Results of clustering for part C - (above) for the k-Means and (below) for the Mean Shift method. (Left) input points, and (right) points coloured by cluster. The tools >20 mm size have 0 machinable volume.

volume while automatically identifying non-usable tools.

6 MACHINABLE VOLUMES - THEORETICAL MODEL

The discretely computed machinable volume and the other features all rely on the realistic nature of data such as the feed rates, and the availability of practical parts. The magnitude of the values obtained may affect the results of the clustering. Parallely, a theoretical model of the variation of machinable volume with the tool size can be created, and tested against many parts. The advantage of a model is of course the unlimited, albeit synthetic, generation of part, tool shapes. The repeated simulation of the model can improve it, if the model is so constructed. So, a model is attempted in 2D presently, though the model should be generalizable to 3D as well. Consider a simple 2D part profile having a pocket with a depression at the pocket's base, similar to part B. Let the bounding box of the part be represented by a quadtree, a simpler hierarchical sub-division method. By definition, the cells intercepting the part profile are subdivided at each level. The nature of quadtrees implies that cells near the boundary get successively refined, while those away from the boundary after certain levels. Here, only the exterior cells are considered, and interior cells of the part profile are ignored

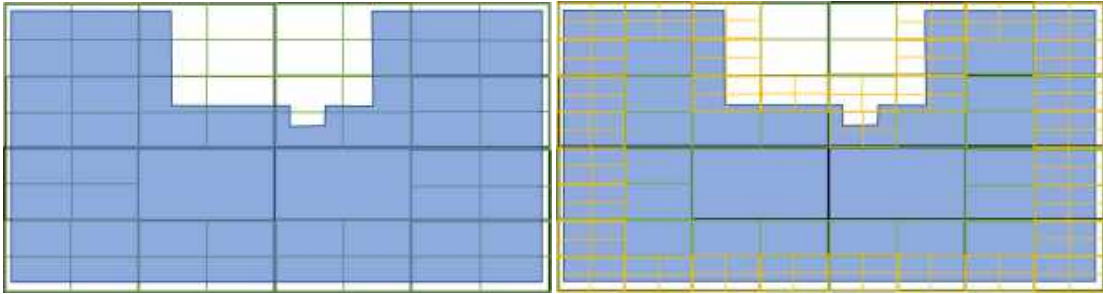


Figure 10: Depiction of accessibility of cells of a quadtrees decomposition - a 2D part (shaded blue) at the (left) second, and (right) third levels of a quadtrees. Cells previously inaccessible become accessible to smaller tools on refinement of the tree.

as they are inaccessible to the tool. The process is shown in Figure 10.

The size of cells can be used to identify the maximum tool size capable of accessing, and machining, a cell; a larger tool would penetrate the part when placed at the said cell. Thus, the set of exterior cells accessible by a tool represents the machinable area for that tool in this 2D setting. The number of machinable cells N_k can be estimated by subtracting the number of boundary cells from cells denoting the total pocket area A_{pocket} as $N_k = A_{pocket}/C_k^2 - L_{pocket}/C_k$; here C_k is the cell size at level k . Further, the number of boundary cells approximated by the length of the pocket's boundary. Thus, the area machinable by the a tool depends on the area of the pocket and the current tool size. The variation is seen to be non-linear in the tool size. It can be plotted for some values of the pocket's parameters.

6.1 Discussion Of The Model

The model developed above approximates the area occupied by boundary cells through the boundary length. However, the length alone may not suffice to estimate the machinable area as boundary curves of the same length may enclose different areas. Also, the interior cells of the target part are not considered here, implying that the machining features are considered as given. An assumption that the area of a shape is same as the area to be machined is possible but not general, as the areas depending on the stock, part shapes among may other factors. The model is thus a starting point to represent trends in machinable areas/volumes with tool sizes, which can be used to 'learn' the estimation of machinable volume with a sufficiently large number of parts ad data.

7 CONCLUSIONS

The use of voxel-based machinable volume computation is simple to implement. The method is general, being applicable to any part and tool shape. The established clustering methods provide a good grouping of the tools. The clusters identify relevant groups reducing the need to exhaustively evaluate all combinations of tools. For the parts considered, the clusters seem to match the intuition of a machinist given the same parts and tools. A suitable combination of criteria/features leads to more realistic clusters than individual criteria. The results of agglomerative and k-means clustering are similar for the parts considered. Nevertheless, agglomerative clustering seems more suited for integration with a CAM system, as the dendrograms render the relation between data points, i.e., tools, more interpretable.

Many avenues of future work open up from the present work. Additional features, such as parameters pertinent to machining, and different methods of clustering can be investigated. The hyperparameter choices for clustering methods need to be explored. The machinable volume computation is extensible to tool rotation,

for e.g. in 5-axis machining. The method should be enhanced for realistic *stock* shapes. Tool *sequences* can be evaluated by through Boolean operations on intermediate machinable volumes. Lastly, the theoretical model for machinable volume should be realized with synthetic and industrial parts in a ‘machine-learning’ approach.

ACKNOWLEDGEMENTS

We thank Manish R. of support, Anand M., product definition for help with machining parameters, Nitin U., Vivek G., Swadhin B. of product management, all from CAMWorks, for problem initiation. HCL Technologies is thanked for providing the facility for working. We appreciate the anonymous reviewers’ comments which helped improve the paper.

Hariharan Krishnamurthy, <http://orcid.org/0000-0002-3059-4132>

Manoj W. Bhonge, <http://orcid.org/0000-0001-7638-4600>

Tathagata Chakraborty, <http://orcid.org/0000-0002-2752-2533>

REFERENCES

- [1] Cluster analysis. Wikipedia. https://en.wikipedia.org/wiki/Cluster_analysis. Last Accessed - 13 April, 2022.
- [2] Akenine-Möller, T.: Fast 3d triangle-box overlap testing. *Journal of graphics tools*, 6(1), 29–33, 2001. <http://doi.org/https://doi.org/10.1145/1198555.1198747>.
- [3] Ameer, A.: Voxel-based tool sequence optimization for 5-axis machining using high performance computing. Master’s thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2017. <https://smartech.gatech.edu/bitstream/handle/1853/60655/AMEUR-THESIS-2017.pdf>.
- [4] Balasubramaniam, M.: Tool selection and path planning in 3-axis rough machining. Master’s thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999. <https://dspace.mit.edu/bitstream/handle/1721.1/80500/43183908-MIT.pdf?sequence=2>.
- [5] Bonaros, B.: K-means elbow method code for python. <https://predictivehacks.com/k-means-elbow-method-code-for-python/>. Last Accessed - 14 April, 2022.
- [6] Bonilla Hernández, A.E.: Analysis and direct optimization of cutting tool utilization in CAM. Ph.D. thesis, University West, Trollhättan, Sweden, 2015. <https://www.diva-portal.org/smash/get/diva2:871936/FULLTEXT02.pdf>.
- [7] Bouaziz, Z.; Zghal, A.: Optimization and selection of cutters for 3d pocket machining. *International Journal of Computer Integrated Manufacturing*, 21(1), 73–88, 2008. <http://doi.org/https://doi.org/10.1080/09511920601164132>.
- [8] Chan, F.T.; Swarnkar, R.: Ant colony optimization approach to a fuzzy goal programming model for a machine tool selection and operation allocation problem in an fms. *Robotics and Computer-Integrated Manufacturing*, 22(4), 353–362, 2006. <http://doi.org/https://doi.org/10.1016/j.rcim.2005.08.001>.
- [9] Churchill, A.W.; Husbands, P.; Philippides, A.: Metaheuristic approaches to tool selection optimisation. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 1079–1086. Philadelphia, PA, USA, 2012. <http://doi.org/https://doi.org/10.1145/2330163.2330313>.
- [10] Company, C.T.: Standard milling calculations. https://conicalendmills.com/wp-content/uploads/2014/10/Standard_Milling_Calculations.pdf. Last Accessed - 14 April, 2022.
- [11] D’Souza, R.; Wright, P.; Sequin, C.: Automated microplanning for 2.5-d pocket machining. *Journal of Manufacturing Systems*, 20(4), 288–296, 2001. [http://doi.org/https://doi.org/10.1016/S0278-6125\(01\)80048-0](http://doi.org/https://doi.org/10.1016/S0278-6125(01)80048-0).

- [12] D'Souza, R.M.: On setup level tool sequence selection for 2.5-d pocket machining. *Robotics and Computer-Integrated Manufacturing*, 22(3), 256–266, 2006. <http://doi.org/https://doi.org/10.1016/j.rcim.2005.06.001>.
- [13] He, Y.; Li, Y.; Wu, T.; Sutherland, J.W.: An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. *Journal of Cleaner Production*, 87, 245–254, 2015. <http://doi.org/https://doi.org/10.1016/j.jclepro.2014.10.006>.
- [14] Hunter, J.D.: Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3), 90–95, 2007. <http://doi.org/https://doi.org/10.1109/MCSE.2007.55>.
- [15] Kyoung, Y.; Cho, K.; Jun, C.: Optimal tool selection for pocket machining in process planning. *Computers & industrial engineering*, 33(3-4), 505–508, 1997. [http://doi.org/https://doi.org/10.1016/S0360-8352\(97\)00179-4](http://doi.org/https://doi.org/10.1016/S0360-8352(97)00179-4).
- [16] 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. <http://doi.org/https://doi.org/10.1115/1.4037631>.
- [17] Nishida, I.; Okumura, R.; Sato, R.; Shirase, K.: Cutting force simulation in minute time resolution for ball end milling under various tool posture. *Journal of Manufacturing Science and Engineering*, 77, 574–577, 2018. <http://doi.org/https://doi.org/10.1016/j.procir.2018.08.218>.
- [18] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; al, e.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830, 2011. <http://doi.org/https://doi.org/10.5555/1953048.2078195>.
- [19] Ramaswami, H.; Shaw, R.S.; Anand, S.: Selection of optimal set of cutting tools for machining of polygonal pockets with islands. *The International Journal of Advanced Manufacturing Technology*, 53(9), 963–977, 2011. <http://doi.org/https://doi.org/10.1007/s00170-010-2909-7>.
- [20] Seth, A.; Stori, J.A.: Optimal tool selection for 2.5 d milling, part 2: a voronoi mountain approach for generalized pocket geometries. *International Journal of Computer Integrated Manufacturing*, 18(6), 463–479, 2005. <http://doi.org/https://doi.org/10.1080/09511920512331319636>.
- [21] Shen, F.; Tarbutton, J.: A voxel based automatic tool path planning approach using scanned data as the stock. *Procedia Manufacturing*, 34, 26–32, 2019. <http://doi.org/https://doi.org/10.1016/j.promfg.2019.06.110>.
- [22] Smid, P.: *CNC programming handbook: a comprehensive guide to practical CNC programming*. Industrial Press Inc., New York, NY, USA, 2007. <https://books.industrialpress.com/9780831133474/cnc-programming-handbook/>.
- [23] Veeramani, D.; Gau, Y.S.: Selection of an optimal set of cutting-tool sizes for 212d pocket machining. *Computer-Aided Design*, 29(12), 869–877, 1997. [http://doi.org/https://doi.org/10.1016/S0010-4485\(97\)00042-0](http://doi.org/https://doi.org/10.1016/S0010-4485(97)00042-0).
- [24] Virtanen, P.; Gommers, R.; Oliphant, T.E.; al, e.: *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. *Nature Methods*, 17, 261–272, 2020. <http://doi.org/https://doi.org/10.1038/s41592-019-0686-2>.
- [25] Wu, L.; Li, C.; Tang, Y.; Yi, Q.: Multi-objective tool sequence optimization in 2.5 d pocket cnc milling for minimizing energy consumption and machining cost. *Procedia Cirp*, 61, 529–534, 2017. <http://doi.org/https://doi.org/10.1016/j.procir.2016.11.188>.
- [26] Yao, Z.; Gupta, S.K.; Nau, D.S.: Algorithms for selecting cutters in multi-part milling problems. *Computer-Aided Design*, 35(9), 825–839, 2003. [http://doi.org/https://doi.org/10.1016/S0010-4485\(02\)00110-0](http://doi.org/https://doi.org/10.1016/S0010-4485(02)00110-0).