MULTIPLE CUTTING TOOL SELECTION IN AUTOMATED PROCESS PLANNING & CNC CODE GENERATION

A Thesis in
Industrial Engineering
by
Bhanu Kishore Jayanthi

© 2014 Bhanu Kishore Jayanthi

Submitted in Partial Fulfillment
of the Requirements
for the Degree of
Master of Science
December 2014
The thesis of Bhanu Kishore Jayanthi was reviewed and approved* by the following:

Christopher Saldana
Assistant Professor of Industrial and Manufacturing Engineering
Thesis adviser

Sanjay Joshi
Professor of Industrial and Manufacturing Engineering

Paul Griffin
Professor of Industrial and Manufacturing Engineering
Head of the Department of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School.
ABSTRACT

Cutting tool selection is one of the important stages in Computer Aided Process Planning (CAPP) that integrates Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). Cutting tool selection is a decision making stage in process planning that provides input to CAM for CNC (Computer Numerically Controlled) code generation. Most of the state-of-art CAM systems still require a human process planning engineer to manually look after catalogues and manuals for selecting appropriate tools for machining a pocket geometry. This consumes a significant amount of time in process planning and hence, cutting tool selection can be considered a bottleneck stage. Also, in order to machine pockets, a process planner has to select more than two tools for optimizing the cutting time and to reduce the tool wear. We consider multiple cutting tool selection problem in this thesis. Since the cutting time is dependent upon several factors like step-over (radial depth of cut), step-down (axial depth of cut), tool geometry, feature geometry, feed rate, cost, etc, multiple cutting tool selection becomes highly constrained and large scaled problem. Using one small tool to machine a feature can eliminate un-machined and gouged areas inside the feature geometry. However, the material removal rate for a small tool is very low and hence consumes a lot of time and incur high production costs. Large tools are known to have higher machine removal rates despite maintaining un-machined areas. Hence, a trade-off or optimization solution is necessary between large and small tools from tools database in order to optimize the fitness function (cost or time taken to manufacture) while machining 2.5-D pocket geometries. In this thesis, feasible tools from the tools library are determined by generating accessible areas of the pocket feature geometries which are inputs from a process planning system. A new approach to open pocket accessible area generation is discussed. The variation of optimal tools sequence by using open pocket approach in place of normal accessible area generation approach is discussed
in the results. Decomposable areas are generated which compute the areas traversed by a tool provided another large tool has already machined the pocket feature. Then optimal tool sequences are determined by using optimization algorithms. Two different optimization routines are explained in this thesis. The first is using Dijkstra’s algorithm which generates exact optimal tool sequences but take a lot of computation time. The second is using genetic algorithm which generates near to optimal solution, however, the computation time is less compared to Dijkstra’s approach. The variation of cutting tool sequences and computation times with these optimization algorithms are presented in the test cases. Also, the variation of tool sequence with different feed rates that impact the cutting tool life has been discussed in the results.
# Table of Contents

List of Figures .......................................................................................................................... vi
List of Tables ............................................................................................................................. vii
ACKNOWLEDGEMENTS ........................................................................................................... viii

1. Introduction ............................................................................................................................. 1
   1.1 Multiple tool selection ....................................................................................................... 2
   1.2 Scope of thesis .................................................................................................................. 4
   1.3 Organization of the thesis ............................................................................................... 4

2. Background ............................................................................................................................. 5
   2.1 Analytical methods .......................................................................................................... 6
   2.2 Geometric algorithms ..................................................................................................... 8
   2.3 Voronoi diagrams ........................................................................................................... 9
   2.4 Offsetting .......................................................................................................................... 12
   2.5 Other Approaches .......................................................................................................... 14

3. Toolpath generation ............................................................................................................... 17
   3.1 Accessible area definition ............................................................................................. 17
   3.2 Tools filtering .................................................................................................................. 20
   3.3 Decomposable area definition ......................................................................................... 21

4. Optimization algorithms ........................................................................................................ 23
   4.1 Dijkstra’s algorithm ....................................................................................................... 23
   4.2 Genetic algorithm .......................................................................................................... 24

5. Results .................................................................................................................................... 27
   5.1 Tool sequence variation with open pocket extension .................................................... 27
   5.2 Tool sequence variation with algorithms ......................................................................... 32
   5.3 Tool sequence variation with fitness function ............................................................... 34

6. Conclusions and future work ............................................................................................... 41
References .................................................................................................................................... 43
List of Figures

Figure 1. Computer aided process planning system................................................................. 2
Figure 2. Graphical representation of possible tool sequences given 3 tools. ............................. 3
Figure 3. Steps to optimal tool sequence generation............................................................... 6
Figure 4. Categories of multiple tool selection ........................................................................... 6
Figure 5. Example of a nested pocket. ....................................................................................... 15
Figure 6. Multiple of possible pockets in a machining setup ...................................................... 16
Figure 7. Examples of different types of open pockets. ............................................................... 16
Figure 8. Examples of pocket features ....................................................................................... 17
Figure 9. Pocket face to closed pocket accessible area generated using tool of radius $r_i$ .......... 18
Figure 10. Closed pocket accessible area calculation steps ....................................................... 18
Figure 11. Pocket face to closed pocket accessible area generated using tool of radius $r_i$ ..... 19
Figure 12. Open pocket accessible area calculation steps ......................................................... 19
Figure 13. Decomposable area generated using tool of radius $r_j$ after using tool of radius $r_i$ . 22
Figure 14. Decomposable area calculation steps ........................................................................ 22
Figure 15. Example of chromosome representation ................................................................. 25
Figure 16. One-point crossover .................................................................................................. 26
Figure 17. Mutation .................................................................................................................... 26
Figure 18. Example 1 of Prototype and the negative volume ...................................................... 27
Figure 19. Accessible area using 1 in and 0.5 in tools ............................................................... 28
Figure 20. Open pocket extension - accessible area using 0.5in tool .......................................... 28
Figure 21. Optimal tool sequences using closed pocket and open pocket approaches ............... 29
Figure 22. Example 2 of prototype and the negative volume ...................................................... 30
Figure 23. Accessible area generated using 0.5 in tool, original approach and open pocket extension. ................................................................. 30
Figure 24. Example 3 of prototype and the negative volume ...................................................... 31
Figure 25. Accessible area generated using 0.275 in tool, original approach and open pocket extension 32
Figure 26. Pocket negative volume ........................................................................................... 33
Figure 27. Accessible areas of the filtered tools for the pocket geometry .................................. 33
Figure 28. Graph showing variation of fitness value (min) with generations ............................ 34
Figure 29. Tool path areas for the optimal tool sequence .......................................................... 34
Figure 30. Graph showing variation of cost with different hourly rate (x/h) values ................. 40
List of Tables

Table 1. Example of a tool database ........................................................................................................... 20
Table 2. Tool sequences and their respective fitness values ........................................................................ 34
Table 3. Tool database (tools having unique feed rates) ........................................................................... 36
Table 4. Tool database (tools with different feed rates and tool life) ......................................................... 38
Table 5. Optimal tool sequence cost by considering only machining cost (fitness function $f(t)$) .......... 39
Table 6. Optimal tool sequence cost by considering machining and tool cost (fitness function $g(t)$). .... 39
Table 7. Variation of tool sequence and cost with tool cost ($x$) to hourly rate ($h$) ratio ......................... 39
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Christopher Saldana for his valuable guidance throughout my studies at Penn State. The discussions with him in the weekly meetings gave me valuable insights on how to approach a solution for complex problems. Though he is a genius and vastly experienced, his down to earth nature gave me the freedom to express my ideas. I would also like to thank him for the continuous support and suggestions in professional life apart from education.

I would like to make a special mention of and Sean P. Turley and Dr. Sanjay Joshi, for their help throughout the course of this thesis. I thank him for all the technical help and suggestions with the modelling software. Dr. Joshi helped me in understanding the problem of this thesis during the weekly ARL meetings. I would also like to thank Dr. Joshi for his valuable time in reading this thesis. I thank my colleague Anuj Datar for his valuable ideas and constructive feedback during difficult situations that helped me complete this thesis in time. I am also thankful to Dr. Paul Griffin for giving me an opportunity to pursue my Masters in Industrial Engineering at Penn State.

Finally, I would like to thank my family for their continuous motivation, moral support and guidance throughout my career.
1. Introduction

Conventional manufacturing using Computer Numerically Controlled (CNC) machines from design inputs requires interpretation of Computer Aided Design (CAD) drawing inputs, map these to machinable features, determine setup directions, generate process plans, and select appropriate tools and machining parameters. This data is fed to Computer Aided Manufacturing (CAM) systems for generation of CNC code. In order to automate this entire process, Computer Aided Process Planning (CAPP) systems which act like a bridge between CAD and CAM systems. Typical steps in a CAPP system include preform selection, setup analysis, volumetric decomposition and feature recognition, cutting tools and machine parameters selection and finally CNC code generation. Preform selection involves interpretation of CAD drawing to derive the size and material of raw material (bounding box). In the setup analysis stage, candidate setup orientations and tool approach directions are analyzed based upon the part geometry. Features like pockets, holes, contours or faces are derived from the part geometry in the volume decomposition stage. Machining parameters are derived from standard libraries in the tool selection stage and in the final stage, the CNC code is generated for each of the machinable features. An example of a typical CAPP system is shown in Fig. 1. Among these stages, cutting tool selection is an important activity which is often accomplished manually in process planning for the following reasons. First, there can be a large number of cutting tools or cutting tool sets for machining a particular pocket geometry. Second, gouging and tolerance issues are to be addressed while selecting a cutting tool. Third, there can be many cutting conditions (machining parameters) for different tool and work piece materials. Due to these challenges in selecting cutting-tools, the user typically uses experience-based judgment and estimates to arrive at a conservative choice of a tool sizes for
machining a given feature. Automating the cutting tool selection process greatly reduces this effort and contributes towards hundred percent automation in process planning.

![Diagram of process planning system](image)

Figure 1. Computer aided process planning (CAPP) system developed by Datar et al. [44]. Steps include: Preform selection, setup analysis, volumetric decomposition, tool selection and CNC code generation. ** Tool selection step is described in this thesis.

### 1.1 Multiple tool selection

Early generation CNC machines required manual loading and set-up of cutting-tools in the spindle. This took considerable amount of time and hence, process planners adopted single tool selection strategy. However with the advent of rapid tool change mechanisms in CNC machines, selection of multiple cutting tools for machining a feature became attractive. Large diameter tools have higher machine removal rates and so can greatly reduce the cutting time, however, their accessibility to tight corners in the features is less. Small diameter tools can reach these corners, however, they have a low machine removal rate and so take longer duration to machine a feature.
The challenge lies in selecting an optimal choice between the tools in the tool library to machine a pocket in the least possible time.

There are many advantages of automating the multiple tool selection stage in process planning systems. First, it greatly reduces the manual effort and time taken to select proper tool sequences, given a feature geometry. Second, it reduces the skilled labor cost associated with selecting appropriate tool sizes. Third, it gives the user a leverage to select tools based on the shop requirements like manufacturing time, manufacturing cost, tool cost, etc.

The multiple cutting tool selection problem in machining a pocket geometry can be graphically represented as shown in the Fig. 2. Suppose there are three tools T1, T2 and T3, where T1 > T2 > T3 with respect to Diameters to machine a pocket geometry. Then the possible sequences are T1→T2→T3, T1→T3, T2→T3 and T3. Total 4 possible sequences as shown in the figure below.

![Graphical representation of possible tool sequences given 3 tools.](image)

Figure 2. Graphical representation of possible tool sequences given 3 tools.

For 3 tools, the number of possible sequences are 4. If there are n tools in the tool library, then the total number of tool sequences possible are $2^{n-1}$. Clearly, we can see that if there are a large number of tools in the tool database (say 10,000), the problem of selecting an optimal sequence from the possible tool sequences becomes very challenging.
1.2 Scope of thesis

Multiple tool selection is a challenging problem that has a lot of scope for future research. The major problems which are addressed in this thesis are:

1) Automating the multiple tool selection process and integrating it to a process planning system by using features (negative volumes) as inputs from CAPP system.

2) Tool screening from the tool library and calculating accessible areas for both closed and open pocket feature geometries.

3) Analysis of tool sequence variation with respect to open pocket extension, different optimization algorithms and fitness functions.

1.3 Organization of the thesis

Chapter 2 deals with the literature review on various tool selection approaches. In Chapter 3, definition of accessible area and decomposable areas are defined along with tool screening. Optimization approaches, Dijkstra’s algorithm and genetic algorithm and their parameters are discussed in Chapter 4. The analysis of tool sequence variation is discussed in detail in Chapter 5. References follow.
2. Background

Cutting tool size determination is important in milling because many factors like tool path length, smooth finishing and machining time depend upon the selected tool. As the cutting tools used in milling have circular cross section, it is impossible to machine sharp corners on the pocket boundary using large diameter tools. Hence, smaller tools are to be used to reach all the areas of the pocket. This resulted in many attempts [1-9] to generate single small tool from a given pocket geometry input. The focus of most of these papers is limited to simple connected geometry and may not be generalized for complicated pocket geometries. Also, the tool paths produced by these methods are not optimal in most cases as they use only one tool to machine the entire area. This is because firstly, small diameter tools can reach smaller corners and gaps but their material removal rates (MRR) are very low. However, large diameter tools have comparatively high MRR but might not be able to access all the corners of a feature. So, there should be a trade-off or a combination of large and small cutting tools to achieve optimal machining time. Secondly, with the introduction of automatic tool-change (ATC) capability to CNC machining centers, we may achieve signification reduction in machining time for a feature by using a set of cutting-tools instead of a single cutting-tool. Thirdly, if multiple tools are used, it becomes necessary to account for other factors such as tool change time and the non-cutting time for the smaller tools to travel between the areas that need to be cut. These factors make the optimization of machining process planning and tool selection a non-trivial process and necessitate the use of many computational methods. Some of the multiple tool selection methods are discussed in the next section.

The first step to multiple tool selection involves determination of the finishing tool that fits the pocket. Often this is limited by the corner radii, narrow corners, narrow gaps between pockets and islands/islands and islands or pocket depth (holder collision). Once a finishing tool is selected,
then a feasible tools list is computed from the available tools along with their tool path lengths using different tool path generation strategies. Most optimization algorithms rely on these tool path lengths for optimization of the cost of machining. Figure 3 below is a representation of a general methodology for optimal tools list generation.

![Figure 3. Steps to optimal tool sequence generation.](image)

The literature on multiple tool selection is extensive. For convenience, this literature has been divided into 5 categories based on the methods used for tool path generation viz., analytical, geometric, Voronoi, offsetting and other approaches. Figure 4 below lists the categories and the sections following it describe each of them in detail.

![Figure 4. Categories of multiple tool selection.](image)

### 2.1 Analytical methods

Analytical methods are the first developed tool path generation approaches. In these methods, the pocket geometries are decomposed into regular shapes and each of these shapes are analyzed using simple geometric algorithms. Veeramani and Gau [1] made the first attempt to solve tool selection problem using analytical approach. They developed an algorithm to select the optimal tool set for machining a triangular pocket using a dynamic programming-based approach.
Analytical models were developed to develop staircase machining strategy and to determine the time for machining the pocket using a given tool. Chen and Zhang [2] plotted the radius graph of the maximum circles along the Medial-Axis Transformation (MAT) of a NURB (non-uniform rational B-Spline) pocket in order to find the largest tool for finish cutting of a free-form pocket without gouging and interference. Then they used particle swarm optimization (PSO) methodology for optimization of cutter size for NURBS curves. However, this method is applicable to the determination of finishing cutters only. For roughing cutters, Chen and Fu [3] used MAT for computing the roughing paths and the area of roughing region, a genetic algorithm (GA) optimization model for finding the multiple largest cutters in terms of the maximum area covered by them is established. However, the above methodologies were unable to handle pocket domains bounded with lines, circular arcs, and B-spline curves. Also they could not identify and eliminate curve/curve intersections, thus computation time was higher.

Recently Chen and Fu [4] proposed a new method to address these issues. They used an optimization model of bisector points, to accurately calculate any bisector point and a hybrid optimization model consisting of PSO and the gradient optimization to efficiently pin-point any bisector point. Thus they achieved higher degree of accuracy and speed in estimating tool path and optimization. Yang et al. [5] decomposed pockets into regular features to best fit multiple cutters of various sizes so that they can efficiently cut the corresponding features without overcut, and then Joneja et al. [6] applied greedy tool planner technique for the above decomposition to different layers of pockets surrounded by sculptured surfaces to determine optimal tool sequence. Ferreira and Ochoa [7] developed a method for generating trochoidal tool paths for 2½ D pocket milling using MAT. The pocket and islands geometry are represented as polygons, and the MAT is calculated as a series of points. The points are then sorted and grouped by an algorithm that
generates the trochoidal path whenever the desired radial depth of cut is attained. In the proposed method, machining time is reduced through a pixel-based simulation, adjusting the tool path to the remaining material.

Analytical methods involve decomposition of pockets into regular shapes to analyze the pocket and derive the tool path. Pocket decomposition results in lower accuracy as they make many approximations when the pocket geometry has many circular contours. So, these methods are generally not accurate in the estimation of tool path. Moreover, the number of decompositions is directly proportional to the complexity of pocket geometry and so, whenever the pocket geometry complexity increases, the computation time increases.

2.2 Geometric algorithms

Geometric algorithms based approaches compute accessible areas of a cutter through geometrical constraints in the pocket geometry. The geometrical constraints such as pocket outer contour and island contour distance, pocket corners, etc are determined using different algorithms. Balasubramaniam et al. [8], observed that milling can be modeled using a network flow formulation. They reduced the computational complexity from an intractable NP-hard problem with $2^n$ possible permutations of n tools, making simplifying assumptions that allow them to sort and cluster tools based on diameter and material removal rate to obtain a near-optimal solution, using Dijkstra’s algorithm to find the shortest path through the network in $O(n^2)$ time. They also extended the concept of machinable area to include the constraints not only of 2D accessibility based on the tool diameter, but also the 3D constraints on tool movement due to tool length, tool holder, and spindle geometry. They showed examples on databases of up to twelve potential tools. Yao et al. [9] proposed an algorithm for determining the largest cutter that can be used to machine a pocket. Building on these results, they developed a methodology for the selection of an optimal
set of cutters for multi-part milling [10]. They describe an algorithm for the computation of accessible and un-machined areas for a cutter based on a modified-offsetting approach in order to avoid problems with traditional offset operators. The selection problem is formulated as a graph search problem and covers multiple parts and cutters. This work was extended to incorporate an exact area computation using an algorithm for computing the open offset [11]. They also addressed the possibility of analyzing parts with multiple 2.5-D features for the target and obstruction profiles and generating an optimal set of tools for multiple such parts. One important limitation of their approach is the approximation of machining time based on areas rather than the actual tool-path length. They considered tool loading and machining time to optimize the total time to manufacture the parts.

These methods like analytical methods depend on the pocket geometry for analyzing the pocket. Hence, with the increase in the complexity of pocket geometry, there will be an increase in the number of constraints to solve the geometry. This results in computational problems. Also, these methods will be able to determine interference free tools, but they necessarily need not be optimal.

### 2.3 Voronoi diagrams

Voronoi diagrams are a way of dividing space into a number of regions. They are a method of partitioning a plane with n points into convex polygons such that each polygon contains exactly one generating point and every point in a given polygon is closer to its generating point than to any other. Voronoi diagrams are used to create tool path trajectories inside the pocket geometry and hence to facilitate multiple tool selection. Held [12] in 1991, was the first to identify the use of Voronoi diagrams in pocket geometries for calculating tool path lengths. Hatna et al. [13] conducted a literature review on pocket machining papers published prior to 1998. In this paper,
the authors showed the relationships among the pocket shape, the cutter, the machine tool, and the cutting conditions in the process of minimizing the cost and the lead time of pocket milling. Voronoi diagrams concept was also introduced to tool-path planning. Veeramani and Gau [14] developed an approach based on the Voronoi mountain (3D Voronoi diagrams) to decompose pockets into sub-pockets to be machined by different tools. They used dynamic programming to select the optimal sequence of tools. However, the methodology does not account for inter sub-pocket boundaries and is restricted to polygonal prismatic pockets with rounded corners only. Inter sub-pocket boundaries are the boundaries between the regions machined by two different tools within a pocket. Details of the construction and properties of the Voronoi mountain concept and the construction of cutter paths considering multiple tool sizes were provided in Veeramani and Gau [15]. However, their approach does not work when there are islands in the pocket, circular pocket edges or when the pockets have open edges (open pockets). Also their methodology was a significant underestimation of the largest feasible tool at each stage due to their treatment of all intermediate edges produced during machining as closed edges. Seth and Stori [16], tried to address all these problems by modifying Voronoi mountains developed by Veeramani and Gau [15], for selecting the sequence of tools to be used to machine a pocket. Their approach is based on identifying critical areas, assigning a value to each tool according to the machining area, and the optimal combination of machining is the one that results in a lower cost. Hinduja and Sandiford [17], developed a procedure to determine two optimal tools with respect to cutting conditions and the tool-path length. They argued that geometrical constraints are equally important to determine optimal tools. They used Voronoi diagrams to determine geometrical constraints, such as the minimum concave radius, the bottleneck width, and the entry distance.
Nadjakova and McMains [18], described an approach using Voronoi diagrams and Dijkstra’s algorithm to find the radii of an optimal set of cutters for machining a pocket with a cutting time within a user specified percentage of that for the optimal cutter radii sequence. Elber et al. [19] provided a scheme, MATHSM, to generate trochoidal tool paths for high-speed machining of free-form pockets, based on their Voronoi diagrams. The central MATHSM toolpath optimizes the radial acceleration exerted on the machining tool, whereas the one-side MATHSM toolpath minimizes the number of retractions-two features that are of extreme importance in high-speed machining. Ramaswami and Anand [20], used polygon decomposition (similar to Voronoi diagrams) to divide any non-convex pocket into convex regions. The tool paths were calculated individually for each of these convex regions and tool size was determined based on a heuristic. Machining is carried out using staircasing strategy. Since the tool size selection was based on a heuristic, the tool path length was not optimal. Ramaswami et al. [21] improvised the algorithm by using graph traversal algorithm to determine the sequence of paths resulting in the lowest path length in order to optimize the rapid traverse time between various machining passes. The methodology included tools retract and approach times to obtain an accuracy estimate of the total processing time. A dynamic programming based approach is used to determine the optimal set of tools for machining the pocket. Mount et al. [22], transformed the multiple milling tool selection problems into a weighted set-covering problem, solved the set-covering problems using a greedy heuristic and produced a logarithmic approximation ratio. Voronoi diagrams were used to discretize the domain (milling boundary) into simple regions and the greedy heuristic is run on these regions.

The Voronoi diagrams based approaches though extensively used in prior research for tool path generation, could not account for pockets with islands. When there are open contours or
circular edges, it becomes difficult to build the initial Voronoi mountain which is used to build tool paths for a particular tool. Also, these methods may not be useful when the pocket geometry is entirely circular (circular pockets). Hence, Voronoi based approaches can best be useful for closed and non-circular boundaries with no islands inside pockets.

2.4 Offsetting

Offsets are generally done on the pocket geometry using CAD software. Usually it involves computing planar offsets of outer pocket contours and inner island contours to determine the feasible area inside a pocket for given tool diameter. Often this area is termed as accessible area. Lim et al. [23] developed a tool ranking scheme for the selection of an optimal tool sequence. Tool accessible area was found using planar offsets of pocket outer geometry and islands which are later combined to produce the accessible region of the pocket using a tool. Later, they optimized the combination of cutting tools based on a set of three-dimensional (3D) volumes or two-dimensional (2D) profiles of the selected machining features [24]. Optimal tools can be selected by considering residual material that is inaccessible to oversized cutters and the relative clearance rates of cutters that can access these regions. However, implementation of this procedure gives no satisfactory results because many machining parameters are not taken into account in the selection process adopted such as the length of the tool path, the cutter feed rate, etc. You et al. [25], [26] developed a quasi-offset algorithm for spiral machining of pockets and subsequent optimal selection of tools based on four hypotheses to aid in the tool selection thereby reducing the computation time required. They proved that no more than four tools are required to machine any pocket geometry. However, the optimal solution achieved using their method is a local optimum bounded by the limit on the number of tools.
D’Souza et al. [27] proposed a graph based methodology in which the tool selection problem was translated into a single sink directed acyclic graph. Accessible areas and decomposable areas are computed using 2D contour offset approach. Dijkstra’s algorithm [28] was used to find the shortest path. D’Souza et al. [29] extended the methodology to finding the tool sequence with the minimum cost of rough machining of free-form pockets on a three-axis milling machine. D’Souza [30], improved upon the graph based methodology by introducing the concept of surrogate tool to accommodate for tool holder collisions. A detailed analysis of graphical representation of multiple tool selection problems with tool holder collisions was done. However, this method is inefficient because it does not evaluate tool pairs between tools that cause holder collisions. Also, graph algorithm requires evaluation of all graph edges (or tool-pairs) for determining optimal tool list from a large tool data set. To overcome these limitations, Ahmad et al. [31] developed a Genetic Algorithm (GA) formulation approach by which the computational effort (number of tool-pairs evaluated) is reduced substantially for large tool sets. Geometric algorithms were developed to find accessible areas and decomposed areas while accounting for tool holders. The methodology was tested between elitist and roulette wheel selection approaches of GA formulation.

Offsetting based approaches are considered to be easiest and cheapest ways to compute optimal tool sequences. This is because offsetting functions are flexible, compatible with any CAD modelling software and can be integrated easily with the commercial process planning systems. Also, the computation cost of generating complex Voronoi mountains or polygons are not present resulting in reduced computation time for optimal tool sequence generation. The computational complexity of these methods rely on the number of offset and Boolean calculations for a given pocket geometry. We will discuss more about this approach in the later parts of the paper.
2.5 Other Approaches

Zhang and Li [32] proposed an algorithm for determining the optimal combination of tools without the need to calculate the tool path. The pocket and island boundaries were polygonized based on a set tolerance, and the machining procedure was simulated using pixel filling. They selected the optimum set of tools based on the hypotheses proposed by You et al. [25]. Li et al. [33] investigated various feasible tool-path patterns for a single tool to cut sculptured parts layer by layer in 2.5-D rough machining. An intelligent approach to automatically identifying the most productive toolpath pattern for a given part layer was introduced. Churchill et al. [34] simultaneously optimized tool sequence and a machining parameter, the cutting speeds of the individual tools through a multiple-tool, multi-objective approach with thickness of excess stock, machining time and tooling costs as objective functions. Unconstrained NSGA-II is used as the base algorithm for computation of the NP hard problem. Tool paths and remaining material information were generated using machining strategist, a tool path generation software.

The tool selection algorithms for pocket features described in this thesis are an extension of offsetting based approach developed by D’Souza et al. [27], [38-40]. The approaches described by D’Souza et al. have many limitations. Firstly, their tool selection methods can use only part geometry data rather than the feature geometry data. Most process planning systems produce negative volumes or features or removal volumes. These features are specifically defined for a particular setup and vary with the setup directions. This creates a problem to integrate the tool selection algorithm with any given process planning system. Secondly, the algorithms developed can be used for a single pocket geometry only. For example, they cannot be applied for nested pockets (see Fig. 5) or multiple pockets in a given setup (see Fig. 6). Thirdly, the methods presented did not account for the generation of feasible tools. The tool libraries in the real world
contain a set of feasible and infeasible tools. Feasible tools are the tools which can remove at least a portion of the pocket geometry or that tools which can generate a finite accessible area (the concept of accessible areas will be discussed in later chapters of the paper). Lastly, as mentioned by Seth & Stori [16], D’Souza et al. accommodated open edges formed during machining but do not address open edges in the pocket boundary. Pockets with open boundaries are called open pockets. Some of the examples are slots, cutouts, steps, etc. (see Fig. 7).

The current approach is an attempt to address all the above problems. We have developed an optimal tool sequence generation method that takes feature geometry input. Our method works well for normal and open pockets. The first stage to our methodology involves generation of feasible tools from the tool library and negative volumes using accessible areas algorithm to reduce computational complexity. In the second stage, tool paths are generated using accessible and decomposable areas for different tool sequences. Finally, optimization algorithms are used to generate optimal tool sequences which reduce the cost or time functions. The subsequent chapters deal with these stages in detail.

Figure 5. Example of a nested pocket.
Figure 6. Example of multiple pockets in a machining setup (Top view).

Figure 7. Examples of different types of open pockets. a) Slot pocket, b) Cutout pocket and c) Step pocket.
3. Toolpath generation

Features generated by a general process planning system are briefly categorized into pockets, faces, contours and holes. While, the faces, contours and holes can be machined tools of large or exact sizes, the real challenge to a planner to select tools lies with pocket features. A pocket is defined as an area with defined borders where material should be milled away. What can be confusing about the term pocket is that the border can be completely or partly outside the material making them invisible to the final part. The pocket geometries which have at least a border-side open is termed as an open pocket and which does not have open edges is termed as a closed pocket. While the definition of pocket feature varies with the process planning system used, the process planning system defined by Datar et al. [35] defined a feature as a closed pocket if all side faces of the single loop are machined faces. If at least one face that is adjacent to the single loop is non-machined it represents an open pocket. Figures 8a and 8b show closed pocket and open pocket features respectively.

![Figure 8. Examples of pocket features. a) Closed pocket, b) Open pocket (one side open). Note that red color face represents machinable area and green color face represents non-machinable area.](image)

3.1 Accessible area definition

Accessible area is defined as the area within the pocket face p that the tool can reach without gouging [30]. This is the area that the tool traverses at every depth of cut to machine
whatever it can in the feature. The algorithm to compute accessible area $A_i$ for a general closed pocket feature face $p$, island faces inside the feature $s$ and tool radius $r_i$ is given below.

C.P.1: Planar offset the outer contour of pocket face $p$ inside by the distance $r_i$.

C.P.2: If islands are present in the pocket feature, planar offset the outer contour of island faces $s$ outside by the distance $r_i$ and combine the islands at their intersections. If there are no islands, move to step C.P.4.

C.P.3: Boolean subtract the volumes in step C.P.2 from the volume generated in step C.P.1.

C.P.4: Planar offset the outer contour of the result in C.P.3 outside by the distance $r_i$ to generate accessible area.

These steps are illustrated for a pocket feature in Fig. 9 and Fig. 10 above. Accessible area for a feature has an important property that the accessible area of a larger diameter tool is a subset of accessible area of a smaller diameter tool. While the above algorithm works for closed pockets (with or without islands), it may not produce valid accessible areas in case of open pockets. The
accessible area for open pockets has to be extended so that the tool traverses over the open-edges
for complete machining. Open pocket features have machinable region m and non-machinable
region n. The following algorithm finds the accessible area of an open pocket feature p with
extended open edges for a given tool t.

O.P.1: Compute the accessible area $A_i$, of the open pocket feature p considering it as closed pocket
i.e., follow steps C.P.1 to C.P.4.

O.P.2: Boolean subtract the area $A_i$ from the open pocket area p. Let us say the Boolean subtraction
result contains areas $S_j$.

O.P.3: Out of these $S_j$ areas, separate the areas that contain machinable region m and non-
machinable region n on the side faces. Call these areas $S_{im}$ and $S_{in}$ respectively.

O.P.4: Now add the area $S_{in}$ to the area $A_i$ to generate the accessible area of the open pocket feature.

![Figure 11. Pocket face to closed pocket accessible area generated using tool of radius ri.](image)

![Figure 12. Open pocket accessible area calculation steps. a) & b) Accessible area calculated using steps C.P.1 to C.P.4 with a tool of radius ri. c) Boolean subtract b from a, separate machinable and non-machinable regions. d) Add non-machinable regions in c to b.](image)
All the steps above are illustrated for an open pocket feature in Fig. 11 and Fig. 12. Though there are many ways to get the output of the open pocket geometry, for convenience with ACIS software [36], we have followed the above methodology.

### 3.2 Tools filtering

Generally, cutting tools used for 2.5-D milling of pockets are flat end mills and ball end mills. We have considered flat end mills for the sake of convenience in our approach. To make the database compatible with the sequence generation algorithms, all the cutting tools in the database are arranged in the descending order of their diameters. The tool parameters are chosen from the machining handbook [37]. Table 1. shows an example of the tools list. Tools are identified by their tool numbers.

<table>
<thead>
<tr>
<th>Tool ID</th>
<th>Tool Diameter (in)</th>
<th>Depth of cut (in)</th>
<th>Width of cut (in)</th>
<th>Feed (in/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>2.25</td>
<td>1</td>
<td>1.125</td>
<td>31</td>
</tr>
<tr>
<td>T2</td>
<td>2</td>
<td>0.9</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>T3</td>
<td>1.75</td>
<td>0.825</td>
<td>0.875</td>
<td>29</td>
</tr>
<tr>
<td>T4</td>
<td>1.625</td>
<td>0.8125</td>
<td>0.8125</td>
<td>28</td>
</tr>
<tr>
<td>T5</td>
<td>1.5</td>
<td>0.75</td>
<td>0.75</td>
<td>27</td>
</tr>
<tr>
<td>T6</td>
<td>1.25</td>
<td>0.6</td>
<td>0.625</td>
<td>26</td>
</tr>
<tr>
<td>T7</td>
<td>1.125</td>
<td>0.55</td>
<td>0.5625</td>
<td>25.5</td>
</tr>
<tr>
<td>T8</td>
<td>1</td>
<td>0.45</td>
<td>0.5</td>
<td>25</td>
</tr>
<tr>
<td>T9</td>
<td>0.875</td>
<td>0.4375</td>
<td>0.4375</td>
<td>24.5</td>
</tr>
<tr>
<td>T10</td>
<td>0.75</td>
<td>0.375</td>
<td>0.375</td>
<td>24</td>
</tr>
<tr>
<td>T11</td>
<td>0.625</td>
<td>0.3125</td>
<td>0.3125</td>
<td>23</td>
</tr>
<tr>
<td>T12</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>22</td>
</tr>
<tr>
<td>T13</td>
<td>0.375</td>
<td>0.175</td>
<td>0.1825</td>
<td>20.5</td>
</tr>
<tr>
<td>T14</td>
<td>0.3125</td>
<td>0.15</td>
<td>0.1562</td>
<td>19.5</td>
</tr>
<tr>
<td>T15</td>
<td>0.275</td>
<td>0.132</td>
<td>0.1385</td>
<td>19</td>
</tr>
<tr>
<td>T16</td>
<td>0.25</td>
<td>0.125</td>
<td>0.125</td>
<td>18</td>
</tr>
<tr>
<td>T17</td>
<td>0.125</td>
<td>0.06</td>
<td>0.0625</td>
<td>15</td>
</tr>
<tr>
<td>T18</td>
<td>0.0625</td>
<td>0.02</td>
<td>0.0312</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 1. Example of a tool database

For any pocket feature, among all the cutting tools in the main tool library, first the critical tools (small and large) are to be identified for machining that particular feature. This is necessary to narrow down the search for the optimal tool sequence of any feature among a list of large (for example, 10,000) tools in the library and hence to reduce the complexity. The accessible areas of
cutting tools are computed starting from the smallest diameter tool in the database. The largest critical tool is the largest cutting tool in the tool database that has non-null accessible area. The smallest critical tool is the largest cutting tool whose accessible area is maximum. In case there are sharp corners in a pocket, the pocket machining is infeasible as no tool can create the sharp corner. However, if the corners need not necessarily be clean for such pockets, the smallest critical tool can be approximated to the smallest tool in the database. Once the largest and smallest critical tools are identified, the tools between and along with them are used in the optimal tool sequence generation algorithms which are discussed in the next chapter in detail.

3.3 Decomposable area definition

Decomposable area is defined as the area with in the pocket face p that a smaller diameter tool machine without gouging after a larger diameter tool machined to the extent of its own accessible area. The decomposable area is the area a smaller tool traverses to create clean corners and to remove empty spaces left after a larger tool has removed a large chunk of material. The algorithm to compute decomposable area $D_j$ of a smaller tool $t_j$, given a pocket feature face p is given below

D.A.1: For the pocket feature p, compute the accessible area $A_i$ using the tool $t_i$ (radius $r_i$) and accessible area $A_j$ using the tool $t_j$ (radius $r_j$).
D.A.2: Planar offset the outer contour of the accessible areas $A_i$ & $A_j$ inside by the distance $r_j$ and inner contours outside by the same distance.
D.A.3: Boolean subtract the result areas of D.A.2 for area $A_i$ from the area $A_j$.
D.A.4: Planar offset the outer contour of the result in D.A.3 outside by the distance $r_j$ to generate decomposable area.
The steps above are illustrated for a pocket feature in Fig. 13 and Fig. 14. The above methodology for computing decomposable area works for closed as well as open pockets. As stated previously, the accessible area of a large cutting tool in a tool sequence is a subset of small cutting tool accessible area. An important property can be deduced from the above statement. If a sequence of tools is used to machine a given pocket, starting with the larger tools first and then the smaller tools, the area that has to be covered by any particular tool is dependent only upon the tool immediately preceding tool in the tool sequence. This property has been exploited in the optimal tool sequence generation algorithms in the next chapter by evaluating the tools sequence as tool pairs.

Figure 13. Pocket face to decomposable area generated using tool of radius $r_j$ after using tool of radius $r_i$.

Figure 14. Decomposable area calculation steps. a) & b): Accessible areas calculated using steps C.P.1 to C.P.4 with tools of radius $r_i$ and radius $r_j$ respectively. c), d), e) & f) Offset outer contour of the pocket inwards and inner contours outside by the tool radii $r_i$ and $r_j$ respectively intersections. g) Boolean subtract f) from e). h) Offset outer contour of g) by $r_j$ outside.
4. Optimization algorithms

The optimization algorithms presented in this chapter minimize the cost (fitness function) of selecting a particular tool sequence for a given pocket feature geometry. All possible tool sequences can be modelled as a path from a single-source, single-sync, directed, acyclic graph (SSDAG). Each node in the graph represents the shape of the pocket after the tool present in the node is done with the machining. In the SSDAG, the start node is the node before the machining commences. The end node is the shape after the smallest critical tool is done machining i.e., the final finishing tool. The weights in the SSDAG represent the cost associated with machining by the tail node of the edge. Since the decomposable area of a tool is dependent upon the tool immediately preceding it in the tool sequence, the computational complexity of solving this SSDAG is dependent upon the number of possible tool pairs in a tools sequence. For a given n tools list, the number of tool pairs possible are $2^n$. However, because of the property stated above, Ahmed et al. [31] showed that the computational complexity of this SSDAG reduces to $0.5*n*(n+1)$. We have used two algorithms to solve the SSDAG. They are Dijkstra’s algorithm (shortest path approach) and genetic algorithm (elitist approach). The next sections deal with these algorithms in detail.

4.1 Dijkstra’s algorithm (Shortest path approach)

Dijkstra’s algorithm [28] is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. For the priority queue of vertexes, using a self-balancing binary search tree, the time bound time complexity of Dijkstra can be reduced to $O(E \log V)$. The distance between edges, $f(t) = \frac{l_m}{r_m} + \frac{l_{air}}{r_{air}} + T_{ch}$, where,
$l_m$ is the machining path lengths obtained from accessible areas and decomposed areas, $l_{air}$ is the air cutting length, $T_{ch}$ is the tool changing time, $f_m$ is machining feed rate of tools and $f_{air}$ is the rapid rate. The optimal tool sequence is nothing but the shortest distance between the start node and the end node (smallest critical tool).

### 4.2 Genetic algorithm

Genetic algorithms are search strategies that employ operations such as recombination, mutation and crossover to determine the class of population that has the best fitness function. As genetic algorithms search the entire population, rather than a single point, they are least prone to getting trapped into a local minimum. The process for genetic algorithm can be summarized as follows:

1. **Initiate Genetic Algorithm**
2. **Generate initial population**
3. **Repeat**
   - Choose two parents for recombination
   - Apply mutation with $P$ probability or
   - Apply crossover with $1-P$ probability
   - Replace parents with offsprings if offsprings are better than parents
   - until stopping condition is reached
4. **Take the best fit chromosome of the final population as the solution.**

The algorithm starts with an initial population of chromosomes through which randomly selected (roulette wheel selection) initial solutions are derived. These solutions are ranked according to fitness functions and the lower/higher fitness valued solutions are eliminated according to the objective function. Next, the screened solutions undergo natural selection processes such as mutation or crossover with respect to probability $P$. This entire cycle is repeated until a termination condition is reached. The genetic algorithm operations has chromosome representation, fitness function, selection, cross-over, mutation and termination. In the chromosome representation stage, tools are represented in a bit string with “1” representing
presence of a tool and “0” representing absence of tool. Since there should always be a critical tool (smallest critical tool), the total bit length of any tool sequence of n tools is n-1. If a tool sequence consists of only T1, T3, T4 and T5, its chromosome representation is as follows in Fig. 15.

```
  1 1 0 1
```

T1 T2 T3 T4

Figure 15. Example of chromosome representation.

The fitness function represents the objective function in the genetic algorithm. In the current implementation, the fitness function is represented by a cost, the cost of machining. It is the sum of costs associated with each tool pair in the tool sequence. Suppose, we want to machine using tool sequence T1→T2→T3→T4, then, the cost of machining is calculated by summing the cost of machining accessible area A1 using tool T1, decomposed area D12 using tool T2 and decomposed area D23 using tool T3.

The fitness function cost in minutes, \( f(t) = \frac{l_m}{f_m} + \frac{l_{air}}{f_{air}} + T_{ch} \), where, \( l_m \) is the machining path lengths obtained from accessible areas and decomposed areas, \( l_{air} \) is the air cutting length, \( T_{ch} \) is the tool changing time, \( f_m \) is machining feed rate of tools and \( f_{air} \) is the rapid rate.

The selection of tool sequences is based on Roulette wheel selection method. In this method, the chromosomes in a population are placed such that the fitness values of each chromosome belong to a scaled interval. Then, a random number is generated and the chromosome corresponding to the interval of the scaled fitness values is the selected tool sequence. Likewise, a second chromosome is selected. These two chromosomes are termed as parent chromosomes. These chromosomes are taken for cross-over or mutation based on probability of cross-over, i.e., if the probability of cross-over is 80%, then, the probability of mutation is 20%. The resulting chromosomes are called children chromosomes. If the fitness values of children are greater than
parents’, then children are selected for next generation, or, else parents are selected. The cross-over operation is done using one-point cross-over rule. In one-point crossover, illustrated in the Fig. 16 below, a crossover site $k$ is selected at random over the string length, and the alleles on one side of the site are exchanged between the individuals. Mutation operation is done so as to ensure that solution does not get stabilized at a local minimum. In this process two random sites are chosen with a probability and their corresponding bits in the bit-strings are reversed. Figure 17 below shows the procedure. The algorithm is terminated after a certain number of fixed generations count. The number of generations can be varied to check the relative accuracy in the solution set.

![Figure 16. One-point crossover.](image16)

![Figure 17. Mutation.](image17)
5. Results

The algorithms developed in this research have been implemented and tested on an Intel Core i7 – 3630 QM, 2.4 GHz processor working on Windows 8.1 operating system platform, coded in C++ programming language. The solid modelling kernel ACIS (spatial technology [38]) was used to generate accessible areas, decomposable areas and for calculating the tool paths. The tool database contains flat end mill cutting tools with their tool ID, depth of cut, width of cut and the feed rate information. Three different test cases are considered for testing the algorithms presented in this paper. The first test case is comparing the tool sequence generated for an open pocket feature using the closed pocket algorithm with that of the extension algorithm for open pockets presented in this paper. The second test case checks the tool sequence variation with respect to different optimization algorithms. The third test case checks the tool sequence variation with respect to a change in the fitness function.

5.1 Tool sequence variation with open pocket extension

Figure 18 shows the prototype and associated negative volume as the first example used for the first study. This prototype geometry contains only the open pocket feature and the process planning system developed by Datar et al. [35], generates the negative volume from the open pocket in Fig. 18.

Figure 18. Example 1 of Prototype and the negative volume (open pocket) along with the dimensions of negative volume to demonstrate open pocket extension.
The tooling library shown in Table 1 is used in this study. It contains 18 different flat end mill (HSS steel) diameter tools along with their machining parameters (width of cut, depth of cut and machining feed rates). In the negative volume shown in the Fig. 18, green faces represent the non-machinable area while red faces represent the machinable area. In filtering the tooling library for feasible tools, the accessible areas are determined and the largest critical tool was found to be the 1 in tool (Fig. 19). Examination of the pocket geometry in Fig. 18 would suggest that the smallest critical tool should be 0.5 in, which is the smallest distance among the corner diameter and the constrictions inside the pocket. The accessible area generated for a 0.5 in tool using the conventional tool selection approach for closed pockets is shown in Fig. 19. From the figure, it can clearly be seen that the corner edges of the negative volume that are adjacent to the open edge of the pocket are not sharp. Thus, the 0.5 in end mill tool cannot be the smallest critical tool in the original approach. This would incorrectly identify the smallest tool in the library as being required to machine the negative volume in Fig. 18, which is in this case a 0.0625 in. The new algorithm for the open pocket extension presented in this thesis was implemented and the modified accessible area for a 0.5 in tool for an open pocket is as shown in Fig. 20.

Figure 19. Accessible area using 1in (largest critical) tool and accessible area using 0.5in tool without using open pocket extension.

Figure 20. Open pocket extension - accessible area using 0.5in (smallest critical) tool.
The output from the modified open pocket algorithm agrees with the theoretically correct result, as the new accessible area for the critical tool is the same as the original open pocket feature area. Using the subset of feasible as inputs to the optimization algorithm (Dijkstra approach), the optimal machining time and corresponding tools (and incremental material removal volumes) in both situations is provided in Fig. 21. Note that in the original tool selection algorithm that machining time is 1.762 min, whereas for the extension it is 0.864 min. The smallest critical tools in these cases are 0.0625 in and 0.5 in, respectively, this yielding the primary differences that was observed in terms of machining time and optimal tooling sequence.

<table>
<thead>
<tr>
<th>Tool diameter (in)</th>
<th>0.5</th>
<th>0.25</th>
<th>0.125</th>
<th>0.0625</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Areas</td>
<td><img src="image-a.png" alt="Accessible Areas" /></td>
<td><img src="image-a.png" alt="Accessible Areas" /></td>
<td><img src="image-a.png" alt="Accessible Areas" /></td>
<td><img src="image-a.png" alt="Accessible Areas" /></td>
</tr>
<tr>
<td>Fitness value (min)</td>
<td>0.86</td>
<td>0.085</td>
<td>0.345</td>
<td>0.472</td>
</tr>
</tbody>
</table>

![Accessible Areas](image-a.png) a

<table>
<thead>
<tr>
<th>Tool diameter (in)</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Areas</td>
<td><img src="image-b.png" alt="Accessible Areas" /></td>
</tr>
<tr>
<td>Fitness value (min)</td>
<td>0.864</td>
</tr>
</tbody>
</table>

![Accessible Areas](image-b.png) b

Figure 21. Optimal tool sequences using closed pocket and open pocket approach to an open pocket feature. a) Sequence generated considering normal pocket approach. Sequence: 0.5 in → 0.25 in → 0.125 in → 0.0625 in. Total time: 1.762 min. b) Optimal Sequence with open pocket extension. Sequence: 0.5 in. Total time: 0.864 min.

Consider the second example shown in Fig. 22 to explain the open pocket extension case. This is an example of a pocket with all sides open. The process planning system by Datar et al. [35], generates the open pocket as shown in Fig. 22. The dimensions of the open pocket feature are also shown in the same figure. We use the same tooling library as in Table 1 to explain this example.
In filtering the tooling library for feasible tools, the accessible areas are determined and the largest critical tool was found to be the 0.875 in tool. Examination of this pocket geometry in Fig. 22 suggests that only constrictions for open pocket geometry should be the distance between walls of pocket (0.433 in) and corner radius towards concave side (radius 0.25 in or diameter 0.5 in) and hence the smallest critical tool should be 0.5 in. Thus, the 0.5 in end mill tool should be the theoretical smallest critical tool using the original approach. So, the accessible area of 0.5 in tool should be the pocket negative volume. However, using the theoretical approach the accessible area for 0.5 in is as shown in Fig. 23 a. It is clear in the figure the accessible area is not enclosing the entire pocket geometry. The original approach gave the next smallest tool in the library (0.375 in) as the smallest critical tool. This resulted in optimal tool sequence of 0.5 in → 0.375 in using the original approach.

Figure 22. Example 2 of prototype and the negative volume (open pocket) along with the dimensions of negative volume to demonstrate open pocket extension.

Figure 23. Accessible area generated by 0.5 in tool using: a) original approach, b) open pocket extension.
Using the new algorithm for the open pocket extension the accessible area for 0.5 in tool is as shown in Fig. 23 b. Since the accessible area of 0.5 in tool is the same as the pocket geometry, the correct optimal sequence is generated during optimization phase as 0.875 in $\rightarrow$ 0.5 in.

Consider the third example shown in Fig. 24. The process planning system by Datar et al. [35], generates the open pocket as shown in Fig. 24. The dimensions of the open pocket feature are also shown in the same figure. Again, we use the same tooling library as in Table 1 to explain this example.

In filtering the tooling library for feasible tools, the accessible areas are determined and the largest critical tool was found to be the 0.625 in tool. Examination of this pocket geometry suggests that the smallest critical tool should be 0.275 in, which can machine the smallest constriction distance (0.309 in) within the pocket geometry. So, the accessible area of 0.275 in tool should be the pocket negative volume. However, using the theoretical approach the accessible area for 0.275 in is as shown in Fig. 25 a. It is clear in the figure the accessible area is not enclosing the entire pocket geometry. The original approach gave the smallest tool in the library (0.0625 in) as the smallest critical tool. This resulted in wrong optimal tool sequence of 0.5 in $\rightarrow$ 0.25 in $\rightarrow$ 0.125 in $\rightarrow$ 0.0625 in using the original approach.
Using the new algorithm for the open pocket extension the accessible area for 0.275 in tool is as shown in Fig. 25 b. Since the accessible area of 0.275 in tool is the same as the pocket geometry, the correct optimal sequence is generated during optimization phase as 0.625 in → 0.275 in.

5.2 Tool sequence variation with algorithms

In the second study of this thesis, the effect of the underlying optimization methodology employed was investigated. To accomplish this, the part geometry of Fig. 26 and the tooling library specified earlier were used as inputs to the optimization study. Two optimization algorithms (Dijkstra, GA), described earlier in Chapter 4, were tested with the candidate part geometry and were compared in terms of optimality of final solution, overall computation time and complexity. Using the modified tool selection algorithm described above, the subset of feasible tools in Table 1 was determined to be restricted to those tools between T5 (1.5 in) and T17 (0.125 in) in the library, as these are the largest and smallest critical tools associated for this part geometry, respectively. The respective accessible areas for the set of feasible tools, shown in Fig. 27, were inputs to the optimization algorithms.
A subset of feasible tool sequences and corresponding fitness values, which are the machining times of each, are shown in the Table 2 below. In the table, the tool sequence is indicated as a binary set of values where ‘1’ represents the presence of a tool in a tool sequence and ‘0’ represents absence of a tool in the sequence. Among the set of feasible tool sequences, the global optimal tool sequence is given by 1.125 in, 0.5 in, 0.25 in and 0.125 in tools, with a corresponding fitness value of 4.72 min. The tool path areas for this sequence are shown in Fig. 28. The GA-based approach was found to be marginally less efficient compared to Dijkstra’s approach. The GA-based approach converged to the optimal value, as shown in the Fig. 28, when the GA ran with an initial population size of 18, 15 generations, probability of mutation 0.05 and probability of cross-over 0.95.
Table 2. Tool sequences and their respective fitness values. The 10th tool sequence is the optimal sequence.

<table>
<thead>
<tr>
<th>Tool Sequence</th>
<th>1.5</th>
<th>1.25</th>
<th>1.125</th>
<th>1</th>
<th>0.875</th>
<th>0.75</th>
<th>0.625</th>
<th>0.5</th>
<th>0.375</th>
<th>0.25</th>
<th>0.275</th>
<th>0.125</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 28. Graph showing variation of fitness value (min) with generations. Optimal value of 4.72 min reached at generation 10.

Figure 29. Tool path areas for the optimal tool sequence (Cutting tools: 1.125 in, 0.5 in, 0.25 in, 0.125 in.).

The convergence of the GA was dependent upon the number of tools in the set of feasible tools, which is dependent upon pocket geometry. The algorithm was re-run 100 times for the same pocket geometry and tools library data to check for the accuracy of the estimation of the optimal
tool sequence and it was found that the GA-approach was able to deliver optimal solutions only 60% of the time, with near-to-optimal solutions 90% of time. This is attributed to the fact that the approach can tend toward local minima for some generations, a well-known issue for most evolutionary based approaches. However, using Dijkstra’s algorithm, the optimal solution was obtained 100% of the time, as the algorithm computes all possible permutations of tool sequences. Thus, it can be concluded that Dijkstra’s approach gives more accurate results than a GA-based approach. The computation time for GA-based approach was found to be 121.76 s, whereas with Dijkstra’s approach the same was computed in 153.58 s. The computation time for solution using the Dijkstra algorithm should be higher than the GA-based approach as all \(0.5*n*(n+1) = 91\) tool pair combinations are computed, compared the GA-based approach which computed only 62 tool pair combinations. Further, it was found that both the approaches yielded similar results when the pocket geometry was simple and when the set of feasible tools is approximately 5.

### 5.3 Tool sequence variation with fitness function

In the third study of this thesis, modifications to the fitness function were made to investigate the effect of tooling parameters on the optimal tool sequence selection process. Tools generally can be used at different feed rates and can have multiple cutting edges. With regard to milling, two flute end mills are known for maximizing the chip clearance, whereas the four flute end mills are known for maximizing the feed rate. Decisions regarding tool feed are largely dependent upon a variety of factors that include the tool material, work piece material and tool coatings. These factors ultimately affect selection of the appropriate cutting conditions as tradeoffs are made between processing rates and tool life, measures that are negatively correlated. The optimal tooling sequence for any given pocket geometry largely depends upon the fitness function, which can balance processing cost (dependent on processing time and labor rate) and tooling cost.
(dependent on tool life and tool cost). It is worthwhile to understand the tradeoffs made in balancing these two variables and the corresponding effects on the optimal tooling sequence.

First as a baseline condition for the study, each tool is given a single set of machining parameters, as in Table 3. While this is the baseline condition for this study, in reality a multitude of machining parameters with infinite resolution are possible for each tool.

<table>
<thead>
<tr>
<th>Tool ID</th>
<th>Tool Dia (in)</th>
<th>Depth of cut (in)</th>
<th>Width of cut (in)</th>
<th>Feed (in/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.5</td>
<td>0.22</td>
<td>0.225</td>
<td>22</td>
</tr>
<tr>
<td>T2</td>
<td>0.45</td>
<td>0.21</td>
<td>0.2125</td>
<td>21.75</td>
</tr>
<tr>
<td>T3</td>
<td>0.425</td>
<td>0.195</td>
<td>0.2</td>
<td>21.5</td>
</tr>
<tr>
<td>T4</td>
<td>0.4</td>
<td>0.18</td>
<td>0.1875</td>
<td>21</td>
</tr>
<tr>
<td>T5</td>
<td>0.375</td>
<td>0.168</td>
<td>0.175</td>
<td>20.5</td>
</tr>
<tr>
<td>T6</td>
<td>0.35</td>
<td>0.156</td>
<td>0.1625</td>
<td>20.25</td>
</tr>
<tr>
<td>T7</td>
<td>0.325</td>
<td>0.144</td>
<td>0.15</td>
<td>20</td>
</tr>
<tr>
<td>T8</td>
<td>0.275</td>
<td>0.132</td>
<td>0.1375</td>
<td>19</td>
</tr>
<tr>
<td>T9</td>
<td>0.25</td>
<td>0.125</td>
<td>0.125</td>
<td>18</td>
</tr>
<tr>
<td>T10</td>
<td>0.225</td>
<td>0.12</td>
<td>0.1125</td>
<td>17</td>
</tr>
<tr>
<td>T11</td>
<td>0.2</td>
<td>0.095</td>
<td>0.1</td>
<td>16</td>
</tr>
<tr>
<td>T12</td>
<td>0.175</td>
<td>0.085</td>
<td>0.0875</td>
<td>15.75</td>
</tr>
<tr>
<td>T13</td>
<td>0.15</td>
<td>0.072</td>
<td>0.075</td>
<td>15.25</td>
</tr>
<tr>
<td>T14</td>
<td>0.125</td>
<td>0.06</td>
<td>0.0625</td>
<td>15</td>
</tr>
<tr>
<td>T15</td>
<td>0.1</td>
<td>0.05</td>
<td>0.05</td>
<td>14.75</td>
</tr>
</tbody>
</table>

Table 3. Tool database (tools having unique feed rates) [31].

The fitness function to be optimized in a case where each tool has a unique processing parameter set is the time to complete the machining process in minutes, \( f(t) = \frac{l_m}{f_m} + \frac{l_{air}}{f_{air}} + T_{ch} \), where, \( l_m \) is the machining path lengths obtained from accessible areas and decomposed areas, \( l_{air} \) is the air cutting length, \( T_{ch} \) is the tool changing time, \( f_m \) is machining feed rate of tools and \( f_{air} \) is the rapid rate. The optimal tool sequence was T1 - 0.5 in (feed rate 22 in/min), T10 - 0.225 in (feed rate 17 in/min) and T14 - 0.125 in (feed rate 15 in/min) with a fitness value of 6.57 min. Assuming an average hourly shop rate of $100, the cost of machining was approximately $10.95 (6.57 min). The breakdown of machining time and machining cost is shown in Table 5.
The baseline case, which uses a unique set of machining conditions for each tool, is now compared to a more realistic scenario wherein a set of multiple possible machining conditions exists for each cutting tool. Generally, it is well understood that faster processing rates, which correspond to more aggressive machining conditions, yield lower tool life. The tool database in Table 4, defined for a HSS tool and 1045 steel work piece, was used wherein 3 different feed rates are given for every tool diameter. For every tool, ‘H’ represents high feed rate, ‘M’ represents moderate feed rate and ‘L’ represents low feed rate. A high feed rate tool has low tool life of 45 min. Moderate feed rate tool has tool life of 90 min. Lowest feed rate tool has tool life of 180 min. These values are taken from machinery handbook ([37]).

<table>
<thead>
<tr>
<th>Tool ID</th>
<th>Tool Dia (in)</th>
<th>Depth of cut (in)</th>
<th>Width of cut (in)</th>
<th>Feed rate (in/min)</th>
<th>Tool life (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (H)</td>
<td>0.5</td>
<td>0.22</td>
<td>0.225</td>
<td>22.00</td>
<td>45</td>
</tr>
<tr>
<td>T1 (M)</td>
<td>0.5</td>
<td>0.22</td>
<td>0.225</td>
<td>18.26</td>
<td>90</td>
</tr>
<tr>
<td>T1 (L)</td>
<td>0.5</td>
<td>0.22</td>
<td>0.225</td>
<td>15.18</td>
<td>180</td>
</tr>
<tr>
<td>T2 (H)</td>
<td>0.45</td>
<td>0.21</td>
<td>0.2125</td>
<td>21.75</td>
<td>45</td>
</tr>
<tr>
<td>T2 (M)</td>
<td>0.45</td>
<td>0.21</td>
<td>0.2125</td>
<td>18.05</td>
<td>90</td>
</tr>
<tr>
<td>T2 (L)</td>
<td>0.45</td>
<td>0.21</td>
<td>0.2125</td>
<td>15.01</td>
<td>180</td>
</tr>
<tr>
<td>T3 (H)</td>
<td>0.425</td>
<td>0.195</td>
<td>0.2</td>
<td>21.50</td>
<td>45</td>
</tr>
<tr>
<td>T3 (M)</td>
<td>0.425</td>
<td>0.195</td>
<td>0.2</td>
<td>17.85</td>
<td>90</td>
</tr>
<tr>
<td>T3 (L)</td>
<td>0.425</td>
<td>0.195</td>
<td>0.2</td>
<td>14.84</td>
<td>180</td>
</tr>
<tr>
<td>T4 (H)</td>
<td>0.4</td>
<td>0.18</td>
<td>0.1875</td>
<td>21.00</td>
<td>45</td>
</tr>
<tr>
<td>T4 (M)</td>
<td>0.4</td>
<td>0.18</td>
<td>0.1875</td>
<td>17.43</td>
<td>90</td>
</tr>
<tr>
<td>T4 (L)</td>
<td>0.4</td>
<td>0.18</td>
<td>0.1875</td>
<td>14.49</td>
<td>180</td>
</tr>
<tr>
<td>T5 (H)</td>
<td>0.375</td>
<td>0.168</td>
<td>0.175</td>
<td>20.50</td>
<td>45</td>
</tr>
<tr>
<td>T5 (M)</td>
<td>0.375</td>
<td>0.168</td>
<td>0.175</td>
<td>17.02</td>
<td>90</td>
</tr>
<tr>
<td>T5 (L)</td>
<td>0.375</td>
<td>0.168</td>
<td>0.175</td>
<td>14.15</td>
<td>180</td>
</tr>
<tr>
<td>T6 (H)</td>
<td>0.35</td>
<td>0.156</td>
<td>0.1625</td>
<td>20.25</td>
<td>45</td>
</tr>
<tr>
<td>T6 (M)</td>
<td>0.35</td>
<td>0.156</td>
<td>0.1625</td>
<td>16.81</td>
<td>90</td>
</tr>
<tr>
<td>T6 (L)</td>
<td>0.35</td>
<td>0.156</td>
<td>0.1625</td>
<td>13.97</td>
<td>180</td>
</tr>
<tr>
<td>T7 (H)</td>
<td>0.325</td>
<td>0.144</td>
<td>0.15</td>
<td>20.00</td>
<td>45</td>
</tr>
<tr>
<td>T7 (M)</td>
<td>0.325</td>
<td>0.144</td>
<td>0.15</td>
<td>16.60</td>
<td>90</td>
</tr>
<tr>
<td>T7 (L)</td>
<td>0.325</td>
<td>0.144</td>
<td>0.15</td>
<td>13.80</td>
<td>180</td>
</tr>
<tr>
<td>T8 (H)</td>
<td>0.275</td>
<td>0.132</td>
<td>0.1375</td>
<td>19.00</td>
<td>45</td>
</tr>
<tr>
<td>T8 (M)</td>
<td>0.275</td>
<td>0.132</td>
<td>0.1375</td>
<td>15.77</td>
<td>90</td>
</tr>
<tr>
<td>T8 (L)</td>
<td>0.275</td>
<td>0.132</td>
<td>0.1375</td>
<td>13.11</td>
<td>180</td>
</tr>
<tr>
<td>T9 (H)</td>
<td>0.25</td>
<td>0.125</td>
<td>0.125</td>
<td>18.00</td>
<td>45</td>
</tr>
<tr>
<td>T9 (M)</td>
<td>0.25</td>
<td>0.125</td>
<td>0.125</td>
<td>14.94</td>
<td>90</td>
</tr>
<tr>
<td>T9 (L)</td>
<td>0.25</td>
<td>0.125</td>
<td>0.125</td>
<td>12.42</td>
<td>180</td>
</tr>
<tr>
<td>T10 (H)</td>
<td>0.225</td>
<td>0.12</td>
<td>0.1125</td>
<td>17.00</td>
<td>45</td>
</tr>
<tr>
<td>T10 (M)</td>
<td>0.225</td>
<td>0.12</td>
<td>0.1125</td>
<td>14.11</td>
<td>90</td>
</tr>
<tr>
<td>T10 (L)</td>
<td>0.225</td>
<td>0.12</td>
<td>0.1125</td>
<td>11.73</td>
<td>180</td>
</tr>
<tr>
<td>T11 (H)</td>
<td>0.2</td>
<td>0.095</td>
<td>0.1</td>
<td>16.00</td>
<td>45</td>
</tr>
<tr>
<td>T11 (M)</td>
<td>0.2</td>
<td>0.095</td>
<td>0.1</td>
<td>13.28</td>
<td>90</td>
</tr>
<tr>
<td>T11 (L)</td>
<td>0.2</td>
<td>0.095</td>
<td>0.1</td>
<td>11.04</td>
<td>180</td>
</tr>
<tr>
<td>T12 (H)</td>
<td>0.175</td>
<td>0.085</td>
<td>0.0875</td>
<td>15.75</td>
<td>45</td>
</tr>
<tr>
<td>T12 (M)</td>
<td>0.175</td>
<td>0.085</td>
<td>0.0875</td>
<td>13.07</td>
<td>90</td>
</tr>
</tbody>
</table>
To incorporate the updated information into the optimization, the fitness function was modified to include effects of tooling cost and tool life in terms of cost ($), \( g(t) = \left( \frac{l_{m}}{f_{m}} + \frac{l_{air}}{f_{air}} + T_{ch} \right) \cdot \frac{h}{60} + \frac{t_{life}}{l_{air}} \cdot x \).

where, \( l_{m} \) is the machining path lengths obtained from accessible areas and decomposed areas, \( l_{air} \) is the air cutting length, \( T_{ch} \) is the tool changing time, \( f_{m} \) is machining feed rate of tools, \( f_{air} \) is the rapid rate, \( h \) is the hourly rate in $/hour, \( x \) is the cost of new tool in $ and \( T_{life} \) is the expected tool life. Assuming an hourly rate of $100 per hour for using a CNC milling machine and average cost of a new tool $30, the optimal tool sequence was T1 (H) - 0.5 in (feed rate 22 in/min), T25 (H) - 0.25 in (feed rate 18 in/min), T34 (H) – 0.175 in (feed rate 15.75 in/min) and T40 (H) - 0.125 in (feed rate 15 in/min). The associated fitness function value was $32.17. Breakup of time and cost of machining and cost of tool for the tools in the sequence is shown in Table 6. The breakdown of machining time and machining cost is shown in Table 5. The total cost of machining was $23.14 (13.55 min), cost of tool (with respect to tool life) was $9.03 and the cost of tool change was $0.56 (0.33 min).

From the above results above, tool life was found to be an important factor in determining the total cost of machining and the optimal tool sequence generated. From the fitness function \( g(t) \), the total cost of machining (\( g(t) \)) is clearly linearly dependent upon both the hourly rate (\( h \)) and the average cost a new tool (\( x \)). The coupled effects of these two variables can be considered
in determining the variation of \( g(t) \) with change in the ratio of these variables (x/h). Table 7 below shows the variation of the cost of machining for different values of x/h. From the table it is evident that as tool cost increases, lower feed rate parameter settings are preferred to compensate for importance of longer tool life. The cost drivers are plotted in Fig. 30 as a function of the x/h ratio. At x/h ratios greater 0.8, machining cost remains constant while tooling costs tends increase linearly. This critical ratio value of x/h likely varies with feature geometry, work materials, tooling library and the available set of machining parameters.

<table>
<thead>
<tr>
<th>Tool ID</th>
<th>Tool Dia (in)</th>
<th>Machining time (min)</th>
<th>Machining cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>0.5</td>
<td>4.51</td>
<td>7.52</td>
</tr>
<tr>
<td>T10</td>
<td>0.225</td>
<td>0.82</td>
<td>1.37</td>
</tr>
<tr>
<td>T14</td>
<td>0.125</td>
<td>1.23</td>
<td>2.06</td>
</tr>
<tr>
<td>Total</td>
<td>6.57</td>
<td>10.95</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Optimal tool sequence cost by considering only machining cost (fitness function \( f(t) \)).

<table>
<thead>
<tr>
<th>Tool ID</th>
<th>Tool Dia (in)</th>
<th>Feed rate (in/min)</th>
<th>Machining time (min)</th>
<th>Machining cost ($)</th>
<th>Tool Cost ($)</th>
<th>Total Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (H)</td>
<td>0.5</td>
<td>22</td>
<td>9.85</td>
<td>16.55</td>
<td>6.56</td>
<td>23.11</td>
</tr>
<tr>
<td>T25 (H)</td>
<td>0.25</td>
<td>18</td>
<td>1.15</td>
<td>2.06</td>
<td>0.77</td>
<td>2.83</td>
</tr>
<tr>
<td>T34 (H)</td>
<td>0.175</td>
<td>15.75</td>
<td>1.01</td>
<td>1.83</td>
<td>0.68</td>
<td>2.51</td>
</tr>
<tr>
<td>T40 (H)</td>
<td>0.125</td>
<td>15</td>
<td>1.54</td>
<td>2.7</td>
<td>1.02</td>
<td>3.72</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>13.55</td>
<td>23.14</td>
<td>9.03</td>
<td>32.17</td>
</tr>
</tbody>
</table>

Table 6. Optimal tool sequence cost by considering machining and tool cost (x/h = 0.3 and fitness function \( g(t) \)).

<table>
<thead>
<tr>
<th>x/h</th>
<th>Tool Sequence</th>
<th>Machining cost ($)</th>
<th>Tool Cost ($)</th>
<th>Total Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>T1 (H), T28 (H), T40 (H)</td>
<td>23.12</td>
<td>0.3</td>
<td>23.42</td>
</tr>
<tr>
<td>0.1</td>
<td>T1 (H), T28 (H), T40 (H)</td>
<td>23.12</td>
<td>3.03</td>
<td>26.15</td>
</tr>
<tr>
<td>0.15</td>
<td>T1 (H), T25 (H), T34 (H), T40 (H)</td>
<td>23.14</td>
<td>4.51</td>
<td>27.65</td>
</tr>
<tr>
<td>0.25</td>
<td>T1 (H), T25 (H), T34 (H), T40 (H)</td>
<td>23.14</td>
<td>7.52</td>
<td>30.66</td>
</tr>
<tr>
<td>0.3</td>
<td>T1 (H), T25 (H), T34 (H), T40 (H)</td>
<td>23.14</td>
<td>9.03</td>
<td>32.17</td>
</tr>
<tr>
<td>0.4</td>
<td>T2 (M), T26 (M), T35 (M), T41 (M)</td>
<td>27.75</td>
<td>7.26</td>
<td>35.01</td>
</tr>
<tr>
<td>0.45</td>
<td>T2 (M), T26 (M), T35 (M), T41 (M)</td>
<td>27.75</td>
<td>8.16</td>
<td>35.91</td>
</tr>
<tr>
<td>0.5</td>
<td>T2 (M), T26 (M), T35 (M), T41 (M)</td>
<td>27.75</td>
<td>9.07</td>
<td>36.82</td>
</tr>
<tr>
<td>0.6</td>
<td>T2 (M), T26 (M), T35 (M), T41 (M)</td>
<td>27.75</td>
<td>10.87</td>
<td>38.62</td>
</tr>
<tr>
<td>0.75</td>
<td>T2 (M), T26 (M), T35 (M), T41 (M)</td>
<td>27.75</td>
<td>13.6</td>
<td>41.35</td>
</tr>
<tr>
<td>0.85</td>
<td>T3 (L), T27 (L), T36 (L), T42 (L)</td>
<td>33.28</td>
<td>9.27</td>
<td>42.55</td>
</tr>
<tr>
<td>0.9</td>
<td>T3 (L), T27 (L), T36 (L), T42 (L)</td>
<td>33.28</td>
<td>9.81</td>
<td>43.09</td>
</tr>
<tr>
<td>1</td>
<td>T3 (L), T27 (L), T36 (L), T42 (L)</td>
<td>33.28</td>
<td>10.92</td>
<td>44.22</td>
</tr>
<tr>
<td>1.5</td>
<td>T3 (L), T27 (L), T36 (L), T42 (L)</td>
<td>33.28</td>
<td>16.35</td>
<td>49.63</td>
</tr>
<tr>
<td>2</td>
<td>T3 (L), T27 (L), T36 (L), T42 (L)</td>
<td>33.28</td>
<td>21.81</td>
<td>55.09</td>
</tr>
<tr>
<td>10</td>
<td>T3 (L), T27 (L), T36 (L), T42 (L)</td>
<td>33.28</td>
<td>109.06</td>
<td>142.34</td>
</tr>
</tbody>
</table>

Table 7. Variation of tool sequence and cost with tool cost (x) to hourly rate (h) ratio.
Figure 30. Graph showing variation of machining cost, tool cost and the total cost in $ of manufacturing with different cost of tool to hourly rate (x/h) values.
6. Conclusions and future work

Cutting tool selection is a major activity required in any automated process planning system. State-of-art CAPP systems that integrate CAD and CAM systems often still require a human process planner to select a cutting tool as well as process parameter settings. The requirement to select a sequence of multiple tools is further complicated by the fact that a large number of tools can be used to machine the same feature and a range of machining parameters can be used for each of these tools. In this thesis, a new method based on geometric offsetting was developed to address optimal selection of cutting tools for a part geometry. The new method utilizes pocket feature input from a process planning system as well as an available tooling library to generate optimal tool sequences. Accessible areas are computed for a feature and the tool library is filtered to generate a list of feasible tools and decomposable areas are used in computing the area traversed by previous tools. This method is capable of solving the tool selection problem for open pockets (slot, cut out and step), a problem which prior methods failed to address. With regard to limitations, the algorithms presented in this thesis are applicable to tool selection for 2.5-D pocket geometries with planar base surfaces only. In reality, pocket features with sculptured surfaces are possible and must be addressed by extension of the algorithms developed in this study to variable z-depth features. The major hurdle to sculptured surfaces lies with the current limitation of ACIS technology to address convex cusps generated during accessible area generation. Further, feature – feature interactions are not considered in the present methodology and can pose an issue for situations with interacting or closely-interacting pocket features. Finally, the algorithms presented in this thesis are dependent upon the efficiency of the CAPP system to generate negative volumes. The effects of using either of two different optimization routines were described, one based on the method described by Dijkstra and also a GA-based formulation. It was found that that
the Dijkstra-based optimization approach was slower in computation time but more accurate in generating an optimal solution when compared to the GA-based optimization. It has also been shown that using different fitness functions, the optimal tool sequence for a given feature geometry varies depending on both hourly cost and tooling cost.

Future research might be focused on using algorithms that do not compute all $0.5{n^2} + n$ tool pairs for evaluating decomposable areas while generating optimal tool sequence so to reduce computation time. This requires research on new optimization algorithms that can compute less than $0.5{n^2} + n$ tool pair combinations for evaluating decomposable areas. Also, since the tool sequences are dependent upon the fitness function used, the fitness function can be made to incorporate the amount of remaining volume and tool life in a multi-objective function approach. Finally, tool holder collisions might be taken into consideration in future research to optimal tool sequence generation.
References


