Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Computer-Integrated Manufacturing 22 (2006) 256–266
Robotics and Computer-Integrated Manufacturing
www.elsevier.com/locate/rcim

On setup level tool sequence selection for 2.5-D pocket machining

Roshan M. D'Souza

Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA

Received 24 May 2005; received in revised form 24 May 2005; accepted 13 June 2005

Abstract

This paper describes algorithms for efficiently machining an entire setup. Previously, the author developed a graph based algorithm to find the optimal tool sequence for machining a single 2.5-axis pocket. This paper extends this algorithm for finding an efficient tool sequence to machine an entire setup. A setup consists of a set of features with precedence constraints, that are machined when the stock is clamped in a particular orientation. The precedence constraints between the features primarily result from nesting of some features within others. Four extensions to the basic graph algorithm are investigated in this research. The first method finds optimal tool sequences on a feature by feature basis. This is a local optimization method that does not consider inter feature tool-path interactions. The second method uses a composite graph for finding an efficient tool sequence for the entire setup. The constrained graph and subgraph approaches have been developed for situations where different features in the setup have distinct critical tools. It is found that the first two methods can produce erroneous results which can lead to machine crashes and incomplete machining. Illustrative examples have been generated for each method.

Keywords: CAPP; Tool sequence selection

1. Introduction

Process planning for milling consists of three main tasks. The first identifies removal volumes/machining features/pockets and various access directions to machine them [1–3]. The second clusters them into setups based on the feasibility of machining these removal volumes in a particular direction, and clamping the stock [4,5]. The final task consists of selecting appropriate tool sequences to either minimize machining time or total cost.

Current state of the art process planning systems [6,7] allow users to select 2 or more tools for machining pockets. The actual tool sequence selection is left to the human process planner. The process of time or cost optimization is one of trial and error where complete process planning has to be done in order to validate the plan and calculate costs using NC-Verify systems.

The issue of selecting tool sequences has been addressed by several researchers [8–16]. All these researchers have focused on a single contiguous feature. However, in real life situations, several features are machined in a single setup. Moreover, some of these features may be nested and therefore can have precedence constraints for machining. Tool sequence selection thus becomes a very complex problem because of the various interactions between features in the setup.

A problem similar to the setup level tool sequence optimization has been addressed by Balasubramaniam et al. and Yao et al. [17,18]. Balasubramaniam et al. have developed a graph based tool sequence selection method for rough machining of 3-axis pockets. 3-axis rough machining is converted to a 2.5 axis problem by dividing the pocket into 2.5 D slices. In the graph representation, the nodes represent the tools and the weights of the edges, the cost of machining. The cost of machining is calculated based on the numerical value of the area of the cross sections at each depth of cut of the accessible region and not actual tool-paths. This method

E-mail address: rmdsouza@mtu.edu.

0736-5845/$ - see front matter © 2005 Published by Elsevier Ltd.
doi:10.1016/j.rcim.2005.06.001

Nomenclature

$f(p,h)$	2.5-D feature/pocket represented by an area p and depth h
t_i	end milling cutter with diameter d_i and cutting length l_i
$A_i(f)$	accessible area of tool t_i in feature f. This essentially is the area that t_i traverses at each depth of cut (*doc*) to machine whatever it can in f
$D_{ij}(f)$	area that t_j traverses in feature f at each *doc* after t_i is done machining to the extent of $A(f)_i$
$T_{\text{feas}}(f)$	set of feasible tools to machine f
$T_{\text{opt}}(f)$	set of tools that form the cheapest tool sequence for f
$X(p,h)$	solid obtained by sweeping 2-D area p through distance h
C_{mn}	precedence constraint resulting from f_m nesting in f_n

of cost calculation is grossly inaccurate as it does not account for geometric complexity. It is stated that the shortest path in the graph is the optimal sequence if the numerical value of the accessible areas monotonically increases down the tool-ordering sequence, assuming that the tools are arranged in the decreasing order of diameters. This assumption works only if the machining cost is a function of the numerical value of the machined areas. However, in real life situations, machining cost is a function of tool-path lengths. Tool-paths are generated from the geometry of the accessible regions. Therefore, the graph approach can be used if and only if, for any two tools, the accessible region of the larger tool is a strict subset of the accessible region of a smaller tool. Only then can the weight of any edge in the graph be independent of the path in which it occurs. Yao et al. have formulated a multipart milling problem using the graph approach. The objective is to select a set of tools to be mounted on a machine for machining several distinct parts from several distinct stock pieces. Essentially, this approach is the same as the composite graph approach described in this paper. An unstated assumption in this formulation is that the critical tool for all the parts is the same. In other words, the smallest tool in the available tool set can completely machine each and every pocket in all the parts. We will show that this approach can lead to incomplete machining when this assumption does not hold.

In this paper we have extended the graph based algorithm for selecting the cheapest tool sequence developed earlier [19]. Four approaches were tried out. In the first approach, tool sequence graphs are solved for individual features. Tool-paths for the tools in the resulting tool sequences are connected on a per tool basis to minimize airtime. This is in a sense a local optimization method. This method can lead to tool crashes in certain situations. The latter methods optimize tool sequences by grouping features in sibling levels. The composite tool sequence graph method is similar to the approach adopted by Yao et al. The constrained graph method forces all possible solutions to pass through critical tools. This constraining can lead to sub-optimal solutions. The subgraph method eliminates this constraint while still generating solutions that completely machine the setup.

2. Tool sequence selection for a single pocket

2.1. Accessible area

Accessible area $A_i(f)$ of a tool t_i in a feature $f(p,h)$ (Fig. 1(a)) is the area that the tool t_i traverses at each *doc*, to machine whatever it can without gouging. If p shares an edge with the stock boundary (i.e. there is an *open edge*), then $A_i(f)$ should sufficiently cover the open-edge for complete machining. The area within p that the tool traverses is given by $A(f)_i \cap p$. Smaller tools have larger accessible areas inside the pocket as compared to larger tools. In fact, for any two tools t_i, t_j with diameters $d_i > d_j$, $(A_i(f) \cap p) \subseteq (A_j(f) \cap p)$. The volume that the tool machines inside the pocket is given by $X(A_i(f) \cap p, h)$. In machining this volume, the tool traverses a volume given by $X(A_i(f), h)$ (Fig. 1(b)). If h is larger than the *doc* of the tool, the volume is removed in layers, each of whose thickness is less than or equal to the *doc* of the tool. Appendix A illustrates the algorithm to calculate accessible area.

2.2. Decomposed area

Consider the case where two tools $t_i, t_j : d_i > d_j$ are used to machine the feature $f(p,h)$. The tool t_i will traverse a region given by $A_i(f)$ at each *doc*. Actual

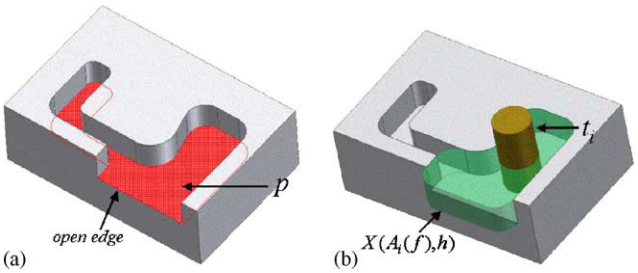

Fig. 1. Finding accessible volume: (a) feature with open-edge, (b) accessible volume given by $X(A_i(f), h)$.

machining is done over the area given by $A_i(f) \cap p$. Tool t_j will machine whatever is left to the extent of $A_j(f) \cap p$. Therefore, the area that t_j machines at each *doc* is nominally given by $A_j(f) \cap p - (A_i(f) \cap p)$ (Fig. 2(a)). However, this area has sharp corners and open-edges (Fig. 2(a)). These open edges occur on the boundary between $A_i(f) \cap p$ and $A_j(f) \cap p$. Therefore, this area has to be extended suitably to generate tool-paths that will completely machine the left over region as well as the open edges. The resulting area is the decomposed area denoted by $D_{ij}(f)$. Also, $(A_i \cap p) \cup D_{ij} = A_j \cap p$. The volume of material removed by t_j is given by $X(A_j(f) \cap p - (A_i(f) \cap p), h)$. In machining this volume, the tool t_j traverses a volume given by $X(D_{ij}(f), h)$ (Fig. 2(b)). Appendix B illustrates the algorithm to calculate decomposed area.

2.3. Finding the feasible tool set

A feasible tool is one which has cutting length greater than the pocket depth, and non-zero accessible area. The feasible tool set for a feature f is given by

$$T_{\text{feas}}(f) = \{t_i : A_i(f) \neq \phi, l_i > h\}. \quad (1)$$

If $T_{\text{feas}}(f) = \phi$, then the feature f is non-manufacturable. Given an available tool set $T = \{t_1, t_2, \ldots, t_n\}$, the procedure for finding the feasible tool set is as follows:

(1) Find the largest tool $t_l \in T$ that can enter the feature f without gouging. This is quickly accomplished using binary partitioning of T. The tool t_l is the largest diameter tool in T for which $A_l(f) \neq \phi$. The set of tools T' that can enter f without gouging is $T' = \{t_i : t_i \in T, d_i \leqslant d_l\}$.
(2) Suppose there exist a precedence constraint C_{mn} between two features f_m,f_n, the largest tool that can enter f_n is smaller than the largest tool than can enter f_m. This is because f_n is nested in f_m.

2.3.1. Finding critical tools

Critical tools are given by $T_{\text{crtc}}(f) = \{t_i : t_i \in T'(f), p - A_i(f) = \phi, l_i \geqslant h\}$. In other words, critical tools are those that can reach everywhere in the pocket without gouging and have enough cutting length to cover the depth of the pocket. Every possible tool sequence will have one critical tool for complete machining.

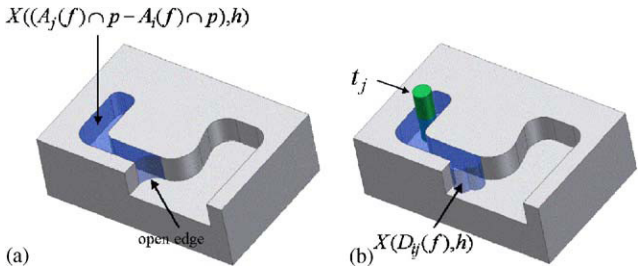

Fig. 2. Finding accessible volume: (a) decomposed volume with open edges, (b) decomposed volume with open edges covered $X(D_{ij}(f), h)$.

The following procedure is used to find the critical tools:

(1) Starting with the smallest tool $t_i \in T'(f)$, Find $A_i(f)$.
(2) If $p - A_i(f) = \phi$, add t_i to $T_{\text{crtc}}(f)$, set $i = i - 1$, go to 1.
(3) If $p - A_i(f) \neq \phi$, set $t_*(f) = t_i$.

The tool $t_*(f)$ is the *smallest non-critical tool* for the pocket f. If $T_{\text{crtc}}(f)$ is an empty set, then the pocket f is non-manufacturable. It makes sense to choose the largest tool in $T_{\text{crtc}}(f)$ as the critical tool $t_c(f)$ for the feature.

Note that the area given by $p - A_*(f)$ can only be machined by the critical tool [20]. When machining this area, the critical tool traverses an area given by $D_{*c}(f)$. If the critical tool cannot traverse this area because of tool holder collision, then the feature is non-manufacturable. The feasible tool set is therefore given by $T_{\text{feas}}(f) = \{t_i : t_i \in T, d_c \leqslant d_i \leqslant d_l, l_i > h\}$.

2.4. Graph algorithm for finding optimal tool sequence

Consider a pocket $f(p, h)$, and a feasible tool set $T_{\text{feas}}(f) = t_1, t_2, \ldots, t_c$. We assume that: (a) larger tools are used before smaller tools, (b) each tool machines whatever it can reach inside f. Let $S_i(f)$ represent the remaining material after t_i is done machining up to the extent of its accessible area. In other words, $S_i(f) = X((p - \{A_i(f) \cap p\}), h)$. If we use two tools t_i, t_j with $d_i > d_j$, the shape of the volume remaining is given by $S_{ij}(f) = X((p - \{A_i(f) \cap p\} - \{A_j(f) \cap p\}), h)$. Clearly, for $d_i > d_j$, $\{A_i(f) \cap p\} \subseteq \{A_j(f) \cap p\}$. In other words, $(A_i(f) \cap p) + (A_j(f) \cap p) = (A_j(f) \cap p)$. This implies that $S_{ij}(f) = S_j(f) = X((p - \{A_j(f) \cap p\}), h)$. In general, $S_{a,b,c\ldots i} = S_i$ for $d_a > d_b .. > d_i$. This means that no matter which larger tool(s) is used before t_j, the shape after t_j is done machining is always the same. If another tool t_k with $d_k < d_j$ was to be used immediately after t_j to machine whatever is left, t_k would always machine the same area at each *doc*.

All possible tool sequences can now be modeled as all possible paths in a graph. The nodes in the graph represent the shape of the pocket after the tool name in the node is done machining. For example node n_i represents the shape $S_i(f)$. This shape is independent of any larger tools used before t_i. The weight of the edge e_{ij} represents the cost of machining the volume $X(D_{ij}(f), h)$ using tool t_j. The start node represents the volume $X(p, h)$. The end node in the graph represents the shape after t_c is done machining ($p - A_c(f) = \phi$). The shortest path in the graph given by Djikstra's algorithm is the optimal tool sequence [21]. The weight of the edge is a function of tool-path lengths, tool life equations, tool

change time, and overhead rate. Job shop specific cost models can be incorporated in the model. Fig. 3 illustrates an example for a feasible tool set containing four tools. Fig. 3(a–e) illustrate the shapes represented by each node in the graph shown in Fig. 3(f).

3. Problem statement

The objective of this research is to find an efficient tool sequence to machine a setup, given a candidate set of tools $T = \{t_1, t_2, \ldots, t_n\}$ with diameters $d_1 > d_2 > \cdots d_n$ and a setup of 2.5-D features $M = \{f_1, f_2 . f_k . f_m\}$ with precedence constraints C_{ik}. The precedence constraints in this research result from feature nesting. Fig. 4 illustrates a typical example. Features f_1, f_2, f_3, f_4 are to be machined when the part is clamped in the orientation as shown. The precedence constraints are C_{14}, C_{23}

4. Method-I: Feature level optimization

In this method, tool sequence graphs are built individually for each feature $f_k \in M$ [19]. This is a local optimization method that does not take into account rapid traversal in air between features, nor the minimization of tool changes across features, for finding the cheapest tool sequence. The weights of the edges of the graph are calculated from the machining time alone. Airtime is a global factor that depends on how individual tool-paths are connected across features. Once the cheapest tool sequence for each feature is found, the individual tool-paths for each tool are connected in such a way as to minimize airtime [22]. While this solution is the cheapest for the individual feature, it may not be the global optimal solution for the entire setup. Machining commences with the largest tool in the set of cheapest tool sequences. The tools used subsequently are in the decreasing order of diameters. For example, consider the cheapest tool sequences $\{T_{\text{opt}}(f_1) = \{t_1, t_3, t_7\}, T_{\text{opt}}(f_2) = \{t_2, t_5\}, T_{\text{opt}}(f_3) = \{t_1, t_2, t_4, t_7\}\}$ for features $\{f_1, f_2, f_3\} \in M$ respectively. The tool diameters are as follows: $d_1 > d_2 > d_3 > d_4 > d_5 > d_7$. Machining commences with t_1, which machines the volumes given by $X(A_1(f_1), h_1) \cup X(A_1(f_3), h_3)$. Subsequently, t_2 is used to machine the volume $X(A_2(f_2), h_2) \cup X(D_{12}(f_3), h_3)$. Finally, t_3 machines the volume $X(D_{13}(f_1), h_1)$. Then t_4 and so on.

4.1. Limitations of the method

The first limitation of this method is because of a false negative non-manufacturable feature condition. The following example illustrates this case. Let f_2 be nested in f_1. Let the cheapest tool sequences for f_1, f_2 be $T^1_{\text{opt}} = \{t_1, t_4\}, T^2_{\text{opt}} = \{t_2, t_3\}$ respectively. The diameters of the tools are as follows $d_1 > d_2 > d_3 > d_4$. When $X(D_{23}(f_2), h_2)$ in f_2 is about to be machined by t_3, f_1

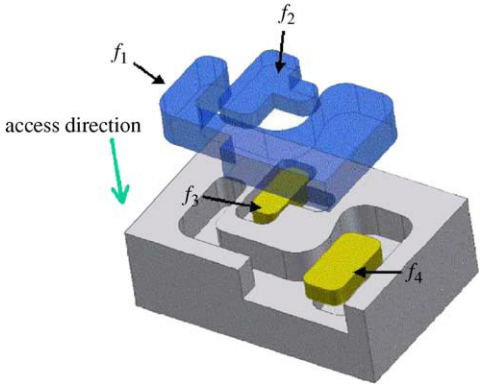

Fig. 4. Feature in a setup.

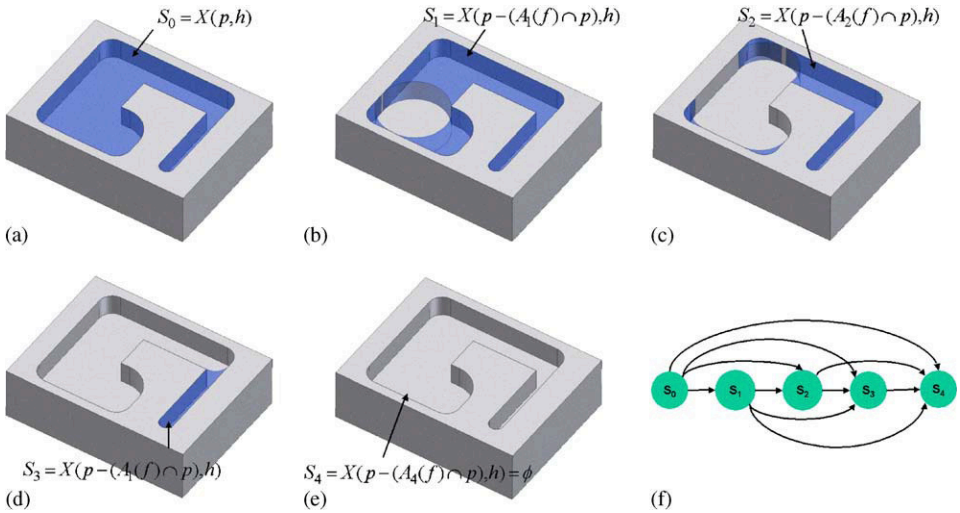

Fig. 3. Graph algorithm formulation: (a) shape S_0, (b) shape S_1, (c) shape S_2, (d) shape S_3, (e) shape S_4, (f) tool sequence graph.

has not been completely machined. Some amount of material given by $X(p_1 - A_1(f_1), h_1)$ is left (Fig. 5(a)). The intermediate stock is therefore given by $I = R - X(A_1(f_1), h_1) - X(A_2(f_2)_3, h_2)$. R represents the shape of the stock before any part of the current setup is machined. The tool holder collision check may show that there is tool holder collision for t_3 while machining f_2 (Fig. 5(a)). However if we adopt an approach of machining where order of machining follows the feature precedence, i.e. f_1 is completely machined, then f_2, the intermediate stock is given by $I = R - X(A_4(f_1), h_1) - X(A_3(f_2), h_2)$. The above mentioned tool holder collision may not occur because there is more clearance for the tool holder (Fig. 5(b)). What this shows is that a feature may be incorrectly rendered non-manufacturable because of the sequence of machining operations that have been chosen. In the latter case, there will be numerous tool changes. For example, if t_3 is part of the efficient tool sequence for both f_1, f_2, since all of f_1 is machined before f_2, a redundant tool change is introduced for t_3.

The second limitation is much more serious in the sense that it can cause machine tool crashes. Since optimal tool sequences are selected on a feature by feature basis, it may so happen that the decomposed area for a tool in a nested feature may be covered by the decomposed area of a smaller tool in the parent feature. For example, if f_2 is nested in f_1, and the optimal tool sequences for f_2 and f_1 are $T_{\text{opt}}(f_1) = t_1, t_3, t_5, t_7$ and $T_{\text{opt}}(f_2) = t_2, t_4, t_7$. Further suppose that $A_5(f_1) - A_3(f_1)$ covers a portion of $A_2(f_2) - A_4(f_2)$, the tool t_4 may have to plunge directly through f_1 to reach f_2. This is because the area $A_5(f_1) - A_3(f_1)$ will not be machined until t_5 is used. Since $d_4 > d_5$, t_5 will only be used after t_4.

5. Sibling level planning

To avoid the problems associated with nesting, we adopt a sibling level planning approach. In this approach, a grouping of features according to the sibling (or generation) level is obtained from the precedence constraints. Cost optimal tool sequence are then selected for each sibling level. All features in a upper sibling level are machined before the lower sibling level. This strategy is explained with the following example. Consider the setup illustrated in Fig. 4. The precedence constraints result in a feature tree as shown in Fig. 6. Features at the same sibling level are grouped together. Here sibling level 1 consists of feature f_1, f_2. Sibling level 2 consists of feature f_3, f_4. Sibling level 1 is completely machined before any part of sibling level 2 is machined.

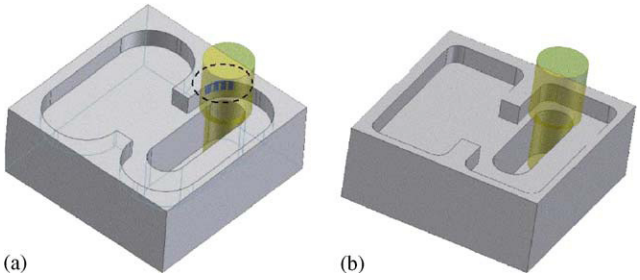

Fig. 5. False negative non-manufacturable condition in feature level optimization: (a) tool holder collision occurs when f_1 is partially machined, (b) tool holder collision does not occur when f_1 is completely machined.

5.1. Method-II: Composite tool sequence graph

In this method, a composite tool sequence graph is generated for all features in a sibling level. The nodes of this graph represent the composite shape of all the features after the tool named in the node is done machining in each feature to the extent of its accessible area. For example, if a sibling level consists of the features f_1, f_2, f_3, the node n_i represents the shape $S_i = X((p_1 - A_i(f_1)), h_1) \cup X((p_2 - A_i(f_2)), h_2) \cup X((p_3 - A_i(f_3)), h_3)$. Similarly, the weight of an edge e_{ij} in the graph is given by the cost of machining the volume $X(D_{ij}(f_1), h_1) \cup X(D_{ij}(f_2), h_2) \cup X(D_{ij}(f_3), h_3)$ using t_j. Airtime for each tool across the different features is accounted for in this method. The graph is built for the union of all feasible tool sets belonging to different features in the setup. The shortest path in the graph is given by Djikstra's algorithm.

5.1.1. Limitations of method-II

Consider a sibling level which consists of features $\{f_1, f_2\}$. Let the feasible tool set for f_1 be $T_{\text{feas}}(f_1) = \{t_1, t_2, t_3, t_4, t_5, t_6\}$. Let t_6 be the critical tool, i.e. the tool that has to be used for complete machining of f_1. Similarly, let the feasible tool set for f_2 be $T_{\text{feas}}(f_2) = \{t_1, t_2, t_3, t_4\}$, with t_4 being the critical tool. The tool sequence graph is constructed for the tools $\{t_1, t_2, t_3, t_4, t_5, t_6\}$. In this graph, the edges $e_{i5}, i = 0, 1, 2, 3, 4$ represent the cost of machining the volumes $X(D_{i5}(f_1), h_1), i = 0, 1, 2, 3, 4$ alone. Similarly, edges

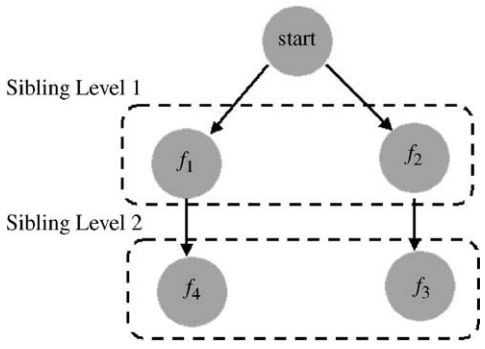

Fig. 6. Sibling levels in a setup.

$e_{i6}, i = 0, 1, 2, 3, 4, 5$ represent the cost of machining the volumes $X(D_{i6}(f_1), h_1), i = 0, 1, 2, 3, 4, 5$. This is because tools t_5, t_6 cannot be used in f_2 as $l_5, l_6 < h_2$, i.e. the tools t_5, t_6 do not have enough cutting length to machine f_2. If the graph were to be solved as such, the cheapest tool sequence could be $\{t_1, t_3, t_6\}$ (Fig. 7(a)). Since t_4 is not part of the cheapest tool sequence, f_2 will not be machined completely.

5.1.2. Tweaking the cheapest tool sequence solution

A solution to this problem is to solve for the shortest path and then tweak the shortest path solution to incorporate all the critical tools. In the example discussed in the previous section, the cheapest tool sequence is $\{t_1, t_3, t_6\}$. This solution is tweaked by adding an additional edge e_{34} (Fig. 7(b)). The weight of this edge is the cost of machining the decomposed feature $X(D_{34}(f_2), h_2)$ alone. This tweaking completes the machining of f_2. Edge e_{34} is chosen because, of all the edges $e_{i4}, i = 1, 3$, e_{34} has least additional cost. Therefore, the solution has effectively two destinations. One is the node representing t_4, and other is the node representing t_6.

A serious problem can occur when the costs are skewed. For example, consider a case when f_1 is small where only the tool t_6 can enter. Let f_2 be large and complex with t_4 as the critical tool. Let $\{0 \rightarrow t_1 \rightarrow t_2 \rightarrow t_4\}$ be the cheapest tool sequence if f_2 were to be solved individually. If the composite graph is solved, the shortest path may be $\{0 \rightarrow t_6\}$. If this solution is tweaked, we would add an additional edge e_{04} which represents the cost of machining f_2 alone using the critical tool. Clearly, this solution is not as good as the solution obtained by solving the individual tool sequence graphs.

5.2. Method III: Constrained graph approach

The constrained graph approach forces all solutions to go through the critical nodes, the nodes representing the shape after critical tools of different features are done machining. Edges that span critical nodes are not considered, thus effectively cutting out paths that will avoid critical tools. In the example discussed in the previous section, all solutions are forced to go through n_4. This effectively splits the graph into two sub-problems. One consisting of the graph for the tools $\{t_1, t_2, t_3, t_4\}$, the other consisting of the graph $\{t_4, t_5, t_6\}$ (Fig. 8). All edges $e_{(4-i)(4+j)}, \{i = 1, 2, 3, 4\}, \{j = 1, 2\}$, that span over the node representing t_4 cannot be considered. In the limiting case, if we have a sibling level consisting of features $\{f_1, f_2, f_3, \ldots, f_n\}$ with critical tools $\{t_1, t_2, t_3, .., t_n\}$ respectively, the constrained graph will consist of the edges $\{e_{01}, e_{12}, e_{23}, e_{34}, e_{45}..e_{(n-1)n}\}$ alone. The cheapest solution in this case will consist of the tool sequence $\{0 \rightarrow t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow \cdots \rightarrow t_n\}$. This could be inefficient as compared to the solution if the features were to be solved individually. This is particularly the case when each feature $f_i, i = 1, 2, 3, 4, 5..n$ can admit all tools $t_j, j = 1, 2 \ldots i$ and has t_i as the critical tool.

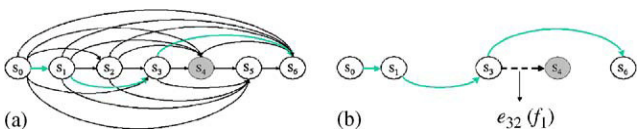

Fig. 7. Composite tool sequence graph: (a) optimal solution does not contain critical tool for f_1, (b) tweaked solution.

5.3. Method IV: Tool sequence sub-graph approach

Consider the example of two features f_1, f_2 in Section 5.2. Let the critical tool for f_1 be t_6, and for f_2 be t_4. Let the tool set be $\{t_1, t_2, t_3 \ldots t_6\}$ with $\{d_1 > d_2 > d_3 \cdots > d_6\}$. We will start building the composite graph where every edge represents the total cost of machining decomposed areas in both features by the tool named in the tail node of the edge. The edge e_{ij} represents the cost of machining $X(D_{ij}(f_1), h_1) \cup X(D_{ij}(f_2), h_2)$ by tool t_j. However, for edges $e_{(4-i)(4+j)}, \{i = 1, 2, 3, 4\}, \{j = 1, 2\}$, that span the critical node n_4, this is not possible as the tool named in the tail node cannot be used in f_2. For example, the edge e_{35} represents the cost of machining decomposed features $X(D_{35}(f_1), h_1) \cup X(D_{35}(f_2), h_2)$ by tool t_5. However, $X(D_{35}(f_2), h_2)$ does not exist as t_5 does not have sufficient cutting length to machine f_2. A slight change in the formulation of the problem solves this predicament. We now assume that for any edge $e_{(4-i)(4+j)}$ spanning the critical node n_4, the tail node $n_{(4+j)}$ represents the shape after all of f_2 is done machining alongwith the shape of f_1 after $t_{(4+j)}$ isdone machining. In other words, $S_{(4+j)} = X(p_2 - (A_4(f_2) \cap p_2), h_2) \cup X(p_1 - (A_{(4+j)}(f_1) \cap p_1), h_1)$. The weight of the edge given by the cost of machining the volume $X(D_{(4-i)(4+j)}(f_1), h_1)$ in f_1, and the minimum cost of completing the machining after tool t_{4-i} is done machining in f_2. This essentially boils down to solving a sub-graph for the cheapest tool sequence between the nodes n_{4-i} and n_4 for feature f_2. For example, the weight of the edge e_{16} (Fig. 9(a)) in the composite graph is the cost of machining $X(D_{16}(f_1), h_1)$ in addition to the cost of the cheapest path of the sub-graph shown in Fig. 9(b) for feature f_2 alone.

Once the composite graph has been built, it can be easily solved for the shortest path to obtain the globally

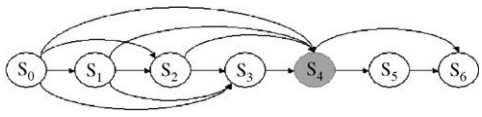

Fig. 8. Constrained graph.

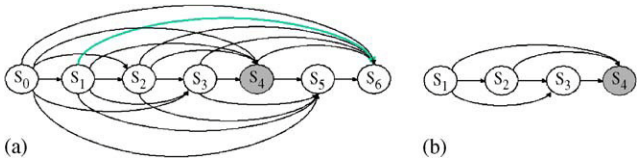

Fig. 9. Sub-graph optimization: (a) edge spanning critical tool t_4 for f_1, (b) sub-graph associated with edge.

cheapest sequence for all the features in the sibling level. If the cheapest solution contains an edge that spans a critical tool, the sub-graph solution has to be added to the globally cheapest sequence.

6. Implementation and examples

The optimization methods described in the previous sections were implemented in a prototype system. The system uses the ACIS [23] solid modeling engine and runs on Windows XP operating system. Execution times are dependent on the complexity of the features. The tool database used in this example is as shown in Table 1.

Tool change time was assumed to be $t_{ch} = 0.083$ min, overhead rate was $h = \$40/\text{h}$, cost of buying and mounting a new tool in the tool magazine was assumed to be $C_T = \$30$. Tool life was assumed to be $t_{lf} = 30$ min. Rapid feed rate was assumed to be 50 in/min. The cost model for calculating edge weight was:

$$W_{ij} = \frac{(t_{mc} + t_{ch} + t_{\text{air}})h}{60} + \frac{t_{mc}}{t_{lf}} C_T, \tag{2}$$

where t_{mc} is machining tool-path time, t_{air} is air-path time. This cost model amortizes the cost of the tool over the usage time of the tool.

The first example illustrates the potential problem of a tool plunging through an excessive depth when feature level optimization strategy is adopted (Method-I). The part to be machined is shown in Fig. 10(a). There are two features f_1 and f_2, with f_2 nesting in f_1. Feature f_1 has a depth of 0.119 in., and feature f_2, a slot of width 0.41 in. has a depth of 0.179 in. The optimal tool sequence for f_1 was found to be $T_{opt}(f_1) = \{t_1, t_7, t_9\}$. The optimal tool sequence for f_2 is $T_{opt}(f_2) = \{t_6\}$. Machining commences with t_1, followed by t_6, t_7, and t_9. Since $A_7(f_1) - A_1(f_1)$ covers a portion of $A_6(f_2)$ (Fig. 10(b)), t_6 will have to plunge through $h_1 + h_2 = 0.228$ in. of material to machine $A_6(f_2)$. This is much larger than the *doc* given in the table. This could potentially damage the machine or snap the tool.

The rest of the section focuses on the comparison between the four extensions to the graph algorithm. The part for these tests is as shown in Fig. 11. Feature f_1 has a depth of 0.5 in. Feature f_2 has a depth of 0.3 in. The critical tool for f_1 is t_8. The critical tool for f_2 is t_{10}.

Table 1
Tool database

Tool name	Tool dia (in)	Cutting length (in)	WOC (in)	DOC (in)	Feed (in/min)	Speed (rpm)
t_1	1.0	2.0	0.5	0.45	30.6	1909
t_2	0.875	1.6	0.4375	0.39375	26.2	2182
t_3	0.75	1.5	0.375	0.3375	25.5	2546
t_4	0.625	1.2	0.3125	0.28125	24.4	3055
t_5	0.5	1.0	0.25	0.225	22.9	3819
t_6	0.375	0.75	0.1875	0.16875	20.4	5092
t_7	0.3125	0.62	0.15625	0.1406	19.6	6111
t_8	0.25	0.55	0.125	0.1125	18.3	7638
t_9	0.201	0.45	0.1	0.0925	17.4	9547
t_{10}	0.125	0.4	0.0625	0.05625	15.27	15277

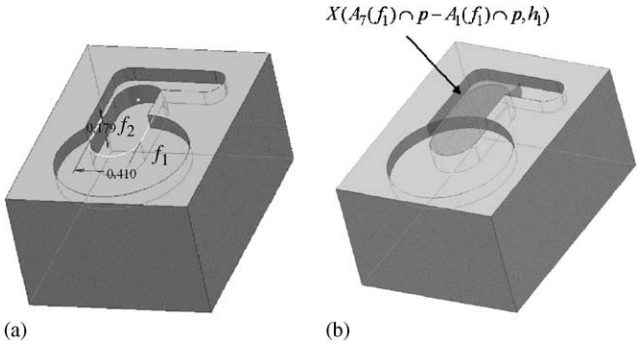

Fig. 10. Potential tool crash from feature level optimization: (a) part, (b) area $A_7(f_1) - A_1(f_1)$ covers $A_6(f_2)$.

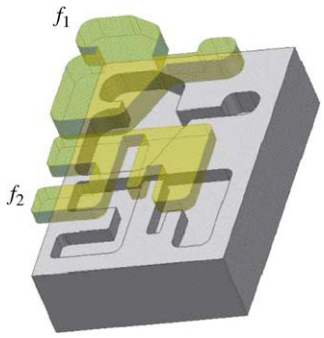

Fig. 11. Example part.

6.1. Feature level optimization

The results of feature level optimization are shown in Table 2. Fig. 12(a–f) show the tool-paths that have been generated. The airtime is minimized across the features for each tool after optimal sequences for each feature have been found. The optimal tool sequence for f_1 is $T_{\text{opt}}(f_1) = \{t_3, t_5, t_8\}$ and for f_2 is $T_{\text{opt}} = \{t_3, t_6, t_9, t_{10}\}$.

6.2. Composite tool sequence graph

Table 3 shows the result of applying Method-II to find the tool sequences. The optimal tool sequence for the

setup is given by $\{t_3 \rightarrow t_6 \rightarrow t_9 \rightarrow t_{10}\}$. Notice that the solution does not contain the critical tool t_8 of feature f_1. Djikstra's algorithm for this graph generated the shortest path without considering completeness of machining. Fig. 13(a–f) shows the tool-paths for each tool. Fig. 13(e) shows the final shape after machining has been completed. Notice that some portion of f_1 is yet to be machined.

6.3. Constrained graph optimization

In the constrained graph method, the original graph was split into two graphs G_1, G_2. Graph G_1 consisted of the nodes $\{n_0, n_1, n_2 \ldots n_8\}$ and graph G_2 consisted of the nodes $\{n_8, n_9, n_{10}\}$. Each edge e_{ij} in G_1 represented the cost of machining $X(D_{ij}(f_1), h_1) \cup X(D_{ij}(f_2), h_2)$. Each edge e_{ij} in G_2 represented the cost of machining $X(D_{ij}(f_2)$ alone. Table 4 shows the results. Fig. 14(a–d) show the tool-paths for each tool in the optimal tool sequence.

Table 2
Machining costs using feature level optimization

Tool name	Tool dia (in)	t_{air} (min)	t_{mc} (min)	Cost ($)
t_3	0.75	0.109129	1.003669	1.745534333
t_5	0.5	0.232829	0.664012	1.317461556
t_6	0.375	0.187729	1.424007	2.554053222
t_8	0.25	0.043491	0.869199	1.533214556
t_9	0.2	0.148409	0.934618	1.712191556
t_{10}	0.125	0.150464	0.838511	1.553383222
			Total cost ($)	10.41583844

6.4. Sub-graph optimization

In sub-graph optimization, there is a global tool sequence graph G to generate the globally optimal tool sequence. For each edge spanning the node representing critical tool t_8 for f_1 in the global graph, a sub-graph is solved to find the optimal sub sequence to complete machining of f_1. Table 5 shows the results of this method. The global tool sequence is $\{t_3 \rightarrow t_6 \rightarrow t_9 \rightarrow t_{10}\}$. In this global sequence, the edge e_{69} spans the node n_8 representing the shape S_8. The optimal sub sequence to complete machining of f_1 is $\{t_6 \rightarrow t_8\}$. Fig. 15(a–e) show the tool-paths for each tool in the global optimal sequence.

7. Conclusions

Tool sequence selection in general is a N–P hard problem. By identifying a structure, namely, the fact that the accessible area of a larger tool is a strict subset of the accessible area of a smaller tool, we were able to reduce the complexity of the problem to $O(n^2)$ for a

Table 3
Machining costs for solution using composite graph approach

Tool name	Tool dia (in)	t_{air} (min)	t_{mc} (min)	Cost ($)
t_3	0.75	0.109129	1.003669	1.745534333
t_6	0.375	0.462344	2.333593	4.253084333
t_9	0.2	0.148409	0.934618	1.712169333
t_{10}	0.125	0.150464	0.838511	1.553383222
			Total cost ($)	9.264149

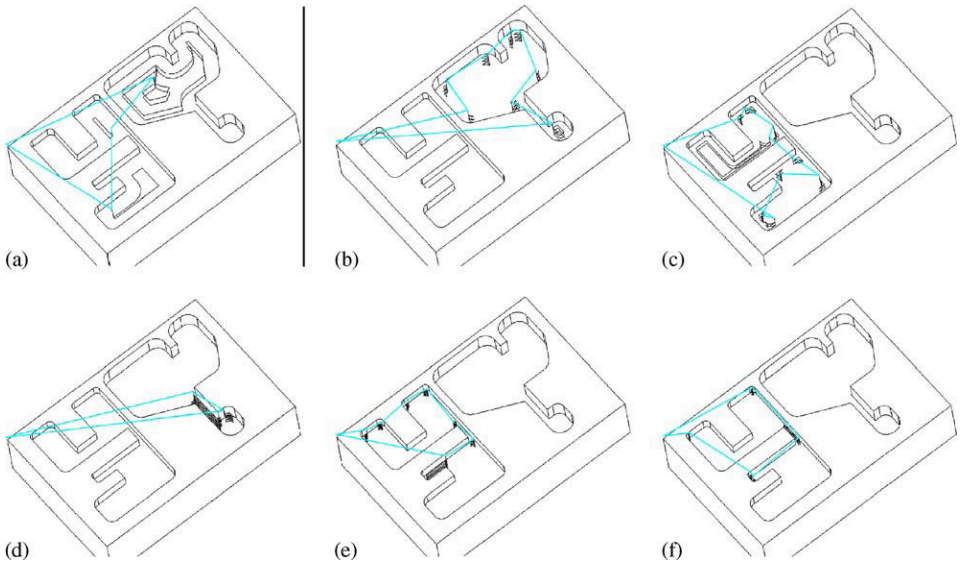

Fig. 12. Feature level tool sequence optimization: (a) tool-paths for t_3, (b) tool-paths for t_5, (c) tool-paths for t_6, (d) tool-paths for t_8, (e) tool-paths for t_9, (f) tool-paths for t_{10}.

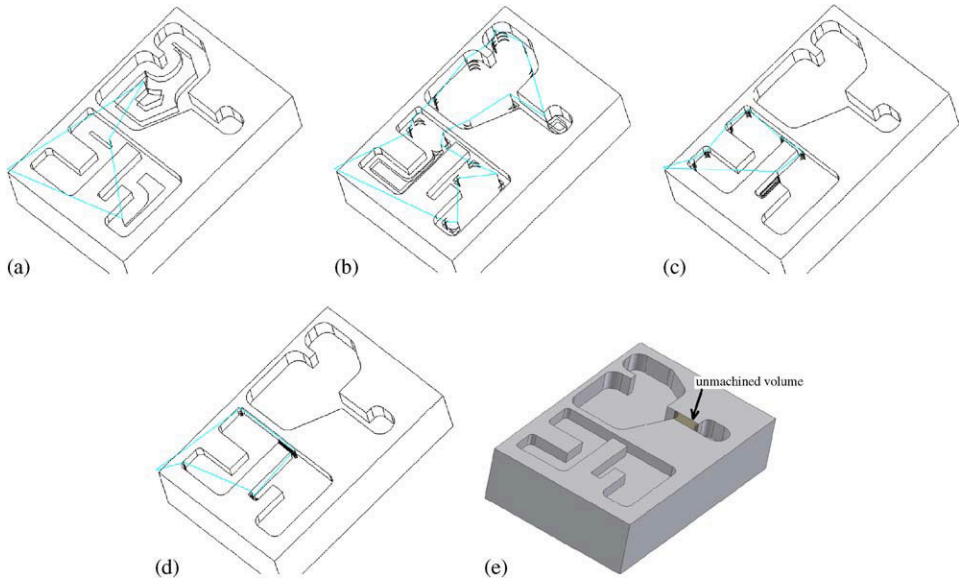

Fig. 13. Optimal tool sequence using composite graph approach: (a) tool-paths for t_3, (b) tool-paths for t_6, (c) tool-paths for t_9, (d) tool-paths for t_{10}, (e) final shape.

Table 4
Machining costs for solution using constrained graph approach

Tool name	Tool dia (in)	t_{air} (min)	t_{mc} (min)	Cost ($)
t_3	0.75	0.109129	1.003669	1.745534333
t_6	0.375	0.462344	2.333593	4.253084333
t_8	0.25	0.199148	1.142871	2.093083667
t_{10}	0.125	0.164425	1.27014	2.28205
			Total cost ($)	10.37375233

Table 5
Machining costs for solution using sub-graph approach

Tool name	Tool dia (in)	t_{air} (min)	t_{mc} (min)	Cost ($)
t_3	0.75	0.109129	1.003669	1.745534333
t_6	0.375	0.462344	2.333593	4.253084333
t_8	0.25	0.012	0.554085	0.987008333
t_9	0.2	0.148409	0.934618	1.712169333
t_{10}	0.125	0.150464	0.838511	1.553361
			Total cost ($)	10.25116

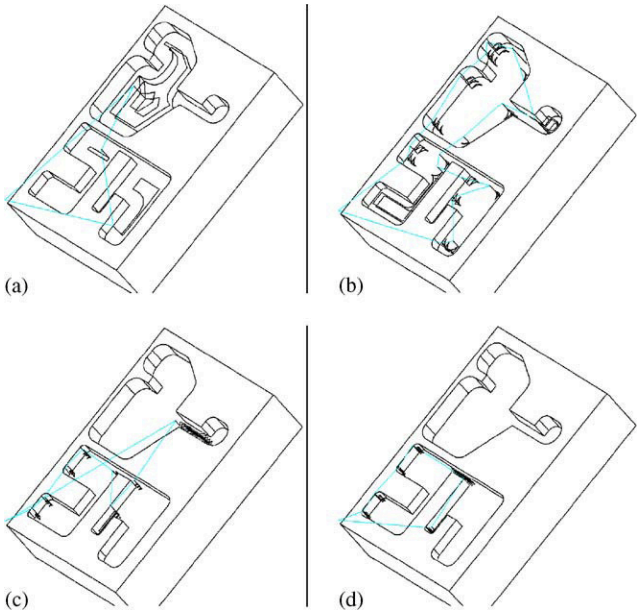

Fig. 14. Optimal tool sequence using constrained graph approach: (a) tool-paths for t_3, (b) tool-paths for t_6, (c) tool-paths for t_8, (d) tool-paths for t_{10}.

single feature. The research presented in this publication extends the method to the setup level while still maintaining the polynomial time complexity. Four extensions to the basic algorithm have been presented. These are: (a) feature level optimization, (b) composite graph method, (c) constrained graph method, and (c) sub-graph method. We have discussed the formulation of each of these methods. A detailed comparison between these methods has been presented using examples. These methods generate the optimal solution subject to the assumptions and constraints imposed.

Our cost calculations are based on actual tool-path generated. Unlike most previous publications, we have detailed and solved several problems that are encountered while using multiple tools for machining. These include the problem of stock boundary open edges, pocket decompositions, and tool-path connections. By solving these problems, we are able to generate actual tool-paths to calculate very accurate costs. Currently our method does not handle tool holders. The unstated assumption is that the tools have enough length, (not flute length) such that the entire tool holder assembly

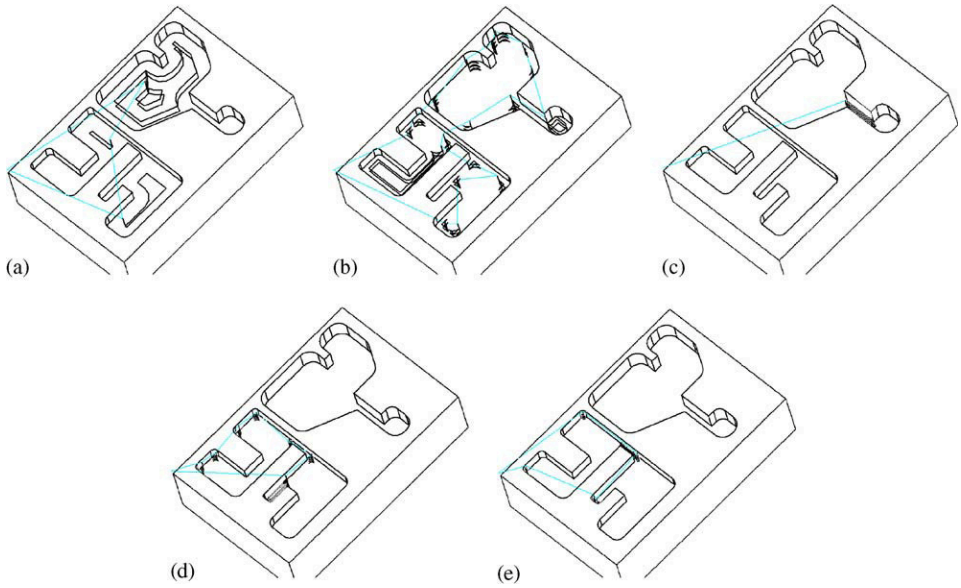

Fig. 15. Optimal tool sequence using sub-graph approach: (a) tool-paths for t_3, (b) tool-paths for t_6, (c) tool-paths for t_8, (d) tool-paths for t_9, (e) tool-paths for t_{10}.

clears the part. The problem structure that we have identified will be lost if this assumption is not true. Future research will address this problem and develop new guided random search methods for finding optimal tool sequences while considering tool holders.

Appendix A. Calculating accessible area $A_i(f)$

The input to the routine that calculates $A_i(f)$, is the tool diameter d_i, the final part P, and the pocket face f

PROCEDURE CALCULATE-ACCESSIBLE-AREA(P, p, d_i)
$W \leftarrow$ Infinite plane at depth h from top plane of the pocket
$X = P \cap W$
$Y = X - p$
$Z = Offset(Y, 0.5d_i)$
$A_i(f) = Offset\ (p - Z, 0.5d_i)$
END

Appendix B. Calculating decomposed area $D_{ij}(f)$

The input to the routine that calculates $D_{ij}(f)$ is the smaller tool diameter d_j, and the accessible areas $A_i(f), A_j(f)$

PROCEDURE CALCULATE-DECOMPOSED-AREA$(A_i(f), A_j(f), d_j)$
$X = Offset(A_i(f), -0.5d_i)$
$Y = Offset(A_j(f), -0.5d_i)$
$D_{ij}(f) = Offset(Y - X, 0.5d_i)$
END

References

[1] Gao S, Shah JJ. Automatic recognition of interacting machining features based on minimal condition subgraph. Computer Aided Design 1998;30(9):695–705.
[2] Regli W. Geometric algorithms for recognition of features from solid models, PhD thesis, University of Maryland at College Park; 1995.
[3] Sundararajan V. Feature recognition and high-level process planning for rapid prototyping using milling, PhD thesis, University of California, Berkeley; 2000.
[4] Echave J, Shah JJ. Automatic setup and fixture planning for 3-axis milling. In: ASME Design Automation Conference, Las Vegas, NV; 1999.
[5] Kannan B, Wright PK. Efficient algorithms for automated process planning of 2.5-d machined parts considering fixturing constraints. Int J Comput Integrated Manufacturing 2004; 17(1):16–28.
[6] MASTERCAM. http://www.mastercam.com, 2002.
[7] SURFCAM. http://www.surfcam.com, 2002.
[8] Bala M, Chang TC. Automatic cutter selection and cutter path generation for prismatic parts. Int J Production Res 1991;29(11): 2163–76.
[9] Arya S, Cheng SW, Mount DM. Approximate algorithm for multiple-tool milling. Int J Comput Geometry and Appl 2001; 11(3):339–72.
[10] Chen Y, Lee YS, Fang SC. Optimal cutter selection and machining plane determination for process planning. J Manufacturing Syst 1998;17(5):371–88.
[11] Lim T, Corney J, Ritchie JM, Clark DER. Optimizing automatic tool selection for 2.5d and 3d nc surface machining. Comput Ind 2000;26(1):41–59.
[12] Kunwoo L, Kim TJ, Hong SE. Generation of toolpath with selection of proper tools for rough cutting. Comput Aided Design 1994;26(11):163–80.
[13] Joo J, Cho H. Efficient sculptured pocket machining using feature extraction and conversion. J Manufacturing Syst 1999;18(2):100–12.
[14] Lee YS, Chang TC. Application of computational geometry in optimization of 2.5 d and 3d nc surface machining. Comput in Ind 1995;26(1):41–59.

[15] Veeramani D, Gau YS. Selection of an optimal set of cutting tool sizes for 2.5d pocket machining. Comput Aided Design 1997;29(12):869–77.
[16] Yao Z, Gupta SK, Nau DS. A geometric algorithm for finding the largest milling cutter. SME J Manufacturing Proc 2001;3(1): 1–16.
[17] Balasubramaniam M, Joshi Y, Engels D, Sarma S, Shaikh Z. Tool selection in three-axis rough machining. Int J Production Res 2001;39(18):4215–38.
[18] Yao Z, Gupta SK, Nau DS. Algorithms for selecting cutters in multi-part milling problems. Comput Aided Design 2003;35: 825–39.
[19] D'Souza R, Wright PK, Séquin CH. Automated microplanning for 2.5d pocket machining. J Manufacturing Syst 2001;20(4):288–96.
[20] D'Souza R, Wright PK, Séquin CH. Handling tool holder collision in optimal tool sequence selection for 2.5-d pocket machining. In: Proceedings of ASME-DETC Computers and Information in Engineering Conference, Montreal, Canada; 2002.
[21] Cormen T, Leiserson C, Rivest R. Introduction to algorithms. McGraw Hill; 1997.
[22] Castelino K, D'Souza R, Wright PK. Tool path optimization for minimizing airtime during machining. J Manufacturing Syst 2003;22(3).
[23] Spatial Technology. Acis Geometric Modeler, API reference, 1994.