Proceedings of CIE' 03
2003 ASME Computers and Information in Engineering
September 2-6, 2003, Chicago,Illinois USA

CIE 2003/CIE-2

SETUP LEVEL TOOL SEQUENCE SELECTION FOR 2.5-D POCKET MACHINING

Roshan M. D'Souza *
Department. of Mechanical Engineering
University of California
Berkeley, California 94720
Email: roshan@me.berkeley.edu

Carlo Séquin
Department of EECS,
Computer Science Division
University of California
Berkeley, California 94720
Email: sequin@cs.berkeley.edu

Paul Wright
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: pwright@robocop.me.berkeley.edu

ABSTRACT

This paper describes algorithms for efficiently machining 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. This work extends earlier research which addressed the issue of selecting an cheapest tool sequence for a single pocket.

NOMENCLATURE

f_a 2.5-D feature represented by 2-D contour p_a and depth h_a.
t_i tool with diameter d_i and cutting length l_i.
A_i^a Accessible area of tool t_i in feature f_a.
D_{ij}^a Region that tool t_j machine in f_a after t_i is done machining to the extent of its accessible area A_i^a.
$T_{feasible}^a$ Set of feasible tools to machine f_a.
T_{opt}^a Set of tools that form the cheapest tool sequence to machine f_a.
$X(p,h)$ Solid obtained by sweeping 2-D contour p along its normal through a distance h.

INTRODUCTION

Process planning for milling consists for three main tasks. The first, identifies removal volumes/machining features and various access directions for machining 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 contigous feature. However, in real life situations, several features are machined in a single setup. Moreover, some of these feature may be nested and can thus have precedence constraints. Tool sequence selection thus becomes a very complex problem because of the various interactions between features in the setup.

In this paper we have extended the graph based algorithm for selecting the cheapest tool sequences developed earlier [17]. Two approaches were tried out. In the first approach, tool sequence graphs are solved for individual features. In the second approach, features are grouped into sibling levels and a composite tool sequence graph is used to find the cheapest tool sequence for all features in a sibling level.

* Address all correspondence to this author.

1 Problem statement

The objective of this research is to find cheapest tool sequence to machine a setup, given a candidate set of tools $T = \{t_1, t_2, ...t_n\}$ with diameters $d_1 > d_2 > .....d_n$ and a setup of 2.5-D features $S = \{f_1, f_2...f_n\}$ with precedence constraints $C_{ij} : f_i \rightarrow f_j$. The precedence constraints in this research result from feature nesting.

2 Finding feasible tool set

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

$$T^a_{feasible} = \{t_i : A^a_i \neq \phi, l_i > h_a\} \quad (1)$$

Note that if the cutting length of a tool is smaller than the pocket depth, then the tool cannot form part of the feasible tool set. If $T^a_{feasible} = \phi$, then the pocket is non-manufacturable. The procedure for finding the feasible tool set is as follows:

1. Find the largest tool $t_i \in T$ that can enter the feature f_a without gouging. This is accomplished using binary partitioning of T. Since T is sorted in the decreasing order of tool diameters, the tools that can enter the feature are given by:

$$T^a_1 = \{t_m : t_m \in T, d_i \geq d_m\} \quad (2)$$

2. Suppose there exist a precedence constraint C_{ab} between two features f_a and f_b, then the feasible tool set for f_b is always a subset of T^a_1, because a tool that cannot enter f_a without gouging, cannot enter f_b without gouging. Repeat 1 to obtain T^b_1 from T^a_1.
3. T^a_1 is sorted by cutting length of the tools. The feasible set of tools is given by:

$$T^a_{feasible} = \{t_x : t_x \in T^a_1, l_x > h_a\} \quad (3)$$

3 Finding criticial tools

Critical tools are defined as those that can cover the entire feature without gouging. The set of critical tools $T^a_{critical}$ for a feature f_a is given by:

$$T^a_{critical} = \{t_x : t_x \in T^a_{feasible}, p_a - A^a_i = \phi\} \quad (4)$$

The following procedure is used to find the critical tools:

1. Starting with the smallest tool $t_i \in T^a_{feasible}$, Find A^a_i.
2. If $p - A^a_i = \phi$, add t_i to $T^a_{critical}$, set $i = i - 1$, go to 1.
3. If $p - A^a_i \neq \phi$, set $t^a_* = t_i$.

The tool t^a_* is the smallest non-critical tool for the feature f_a.

4 Finding cheapest tool sequence: Method-I

In this method, tool sequence graphs are built for each feature $f_a \in S$. This is in a sense a local optimization method that does not take into account interactions between features for finding the cheapest tool sequence. The weights of the edges of the graph are calculated from the machining time alone. Air-time 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 air-time. 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 cheapest tool sequences $\{T^1_{opt} = \{t_1, t_3, t_7\}, T^2_{opt} = \{t_2, t_5\}, T^3_{opt} = \{t_1, t_2, t_4, t_7\}\}$ for features $\{f_1, f_2, f_3\}$ 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 A^1_1, A^3_1. Subsequently, t_2 is used to machine A^2_2, D^3_{12}. D^3_{12} is the decomposed sub-feature that t_2 machines in feature f_3. Then t_3 and so on.

4.1 Planning

Consider a setup $S = \{f_1, f_2..f_n\}$. Let the associated precedence constraints be C_{ij}. The planning starts with the top most feature in a precedence constraint chain and ends at the bottom most feature. For example, if the set of precedence constraints are $\{C_{ab}, C_{bc}\}$, then the order of planning the features is $\{f_a \rightarrow f_b \rightarrow f_c\}$.

The following procedure is used to find the cheapest tool sequence:

1. For each tool $t_i \in T^x_{critical}$, find D^x_{*i}, where t^x_* is the smallest non-critical feasible tool for f_x.
2. If there exist a precedence constraint C_{ix}, then $f_i(p_i, h_i)$ would have been planned. Hence the the cheapest tool sequence T^i_{opt} is already known. The intermediate stock is calculated as:

$$I = R - X(A^i_j, h_i) \quad (5)$$

   R is the intial stock, X is the solid obtained by sweeping the contour through a specified depth. A^i_j is the accessible area of the tool $t_j \in T^i_{opt}$ which is just larger than $t_i \in T^x_{critical}$. The intermediate stock is needed for checking tool holder collision. When tool t_i is about to machine the decomposed sub-feature D^x_{*i}, it can be safely assumed that all tools larger than t_i in the cheapest tool sequences of the features preceeding f_x are done machining.
3. Once the intermediate stock is determined, we can check for tool holder collision for t_i while machining D^x_{*i}. If all

critical tools fail this test then the feature is non manufacturable [18].

4. Next we start building the tool sequence graph by first building the key edges starting with the largest tool in $T^x_{feasible}$. The intermediate stock needed to check for tool holder collision is calculated using the method in step 2. Once key edges have been built, rest of the edges are built without checking for tool holder collision. The graph is solved for the shortest path which is the cheapest tool sequence, and the solution is validated [18].

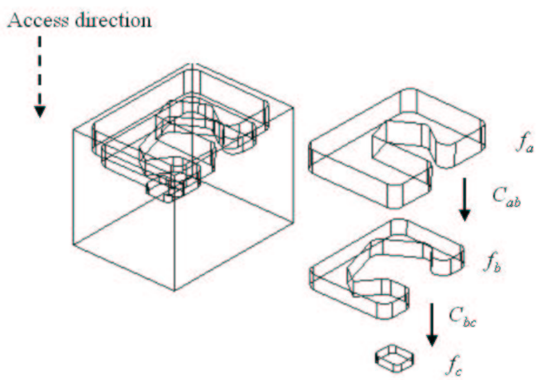

Figure 1. **Feature nesting and precedence constraints**

4.2 Limitations of the method

The method described previously seeks to minimize tool changes while machining. When a certain tool is taken from the tool crousel, it is not returned until all machining assigned to it is completed. For example (Figure 1), if there is a precedence chain given by $\{f_a \rightarrow f_b \rightarrow f_c\}$. If a tool t_i is taken from the crousel, it is used to machine to all region assigned to it in f_a, f_b, f_c. When f_b is being machined by t_i, f_a has not been completely machined. The tool holder collision check may show that there is tool holder collision for t_i while machining f_b if the intermediate stock is calculated using 2 in section 5. However if we adopt an approach of machining where order of machining follows the feature precedence, i.e. f_a is completely machined, then f_b and finally f_c, then the above mentioned tool holder collision may not occur. However, there will be numerous tool change operations.

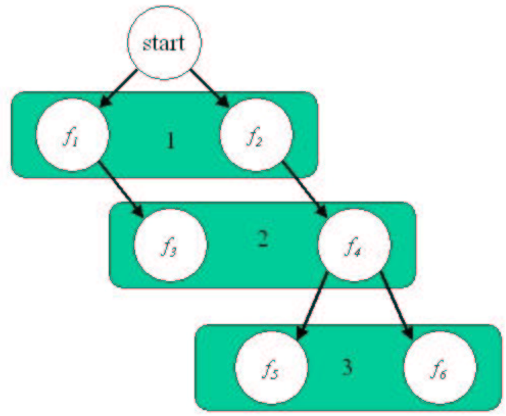

Figure 2. **Feature nesting tree and sibling levels**

5 Finding cheapest tool sequence: Method-II

Consider a setup $S = \{f_1, f_2, f_3, f_4, f_5, f_6\}$. Let the precedence chains be $\{f_1 \rightarrow f_3\}$,$\{f_2 \rightarrow f_4 \rightarrow f_5\}$, and $\{f_4 \rightarrow f_6\}$. The feature tree for this setup is shown in Figure 2. Clearly, the features could be grouped into different sibling levels. In this particular case, sibling level 1 consists of the feature $\{f_1, f_2\}$. Sibling level 2 consists of features $\{f_3, f_4\}$. Sibling level 3 consists of the features $\{f_5, f_6\}$. The sibiling level precedence constraints are therefore $\{f_1, f_2\} \rightarrow \{f_3, f_4\} \rightarrow \{f_5, f_6\}$.

Now we build sibling level tool sequence graphs. For example, in the tool sequence graph $G_1(V, e)$ for sibling level 1, the node $v_i \in V$represents the state of the stock after a given tool t_i is done machining in each of the features (if it can) belonging to the sibling level. Any edge $e_{ij} \in e$ therefore represents the composite cost of machining all the decomposed sub-features $D^a_{ij}, a = 1,2$ in the features $\{f_1, f_2\}$ in sibling level 1. In this method the air-time can be incorporated in the cost. The graph is built for the tools set which is a union of all feasible tool sets belonging to all the features in sibling level 1. The graph can now be solved for the shortest path to get the cheapest tool sequence.

Machining is done on a per sibling level starting from the top most sibling level in the sibling level precedence chain. In above mentioned example, all feature belonging to sibling level 1, $\{f_1, f_2\}$, are completeley machined before machining any part of the features in sibiling level 2. The intermediate stock for sibling level 2 is given by:

$$I = R - X(p_1, h_1) - X(p_2, h_2) \tag{6}$$

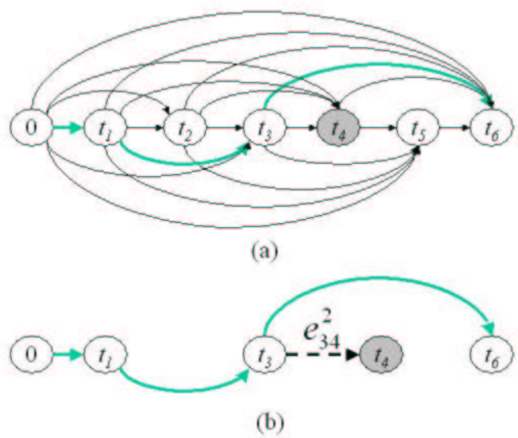

Figure 3. **(a)Shortest path solution does not include t_4 the critical tool for feature f_2. (b) Tweaking the shortest path to include the critical tool**

5.1 Critical tools in solution

Consider sibling level 1 which consists of features $\{f_1, f_2\}$. Let the feasible tool set for f_1 be $T^1_{feasible} = \{t_1, t_2, t_3, t_4, t_5, t_6\}$. 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 $T^2_{feasible} = \{t_1, t_2, t_3, t_4\}$, 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 decomposed feature $D^1_{i5}, i = 0, 1, 2, 3, 4$ alone. Similarly, edges $e_{i6}, i = 0, 1, 2, 3, 4, 5$ reprsent the cost of machining decomposed feature $D^1_{i6}, i = 0, 1, 2, 3, 4, 5$. This is because tools t_5, t_6 cannot be used in f_2 as $l_5, l_6 < h_2$. If the graph were to be solved as such, the cheapest tool sequence could be $\{t_1, t_3, t_6\}$(Figure 3(a)). Since t_4 is not part of the cheapest tool sequence, f_2 will not be machined completely.

A solution to this problem is to force all solutions to go through the node representing t_4(Figure 4(b)). 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

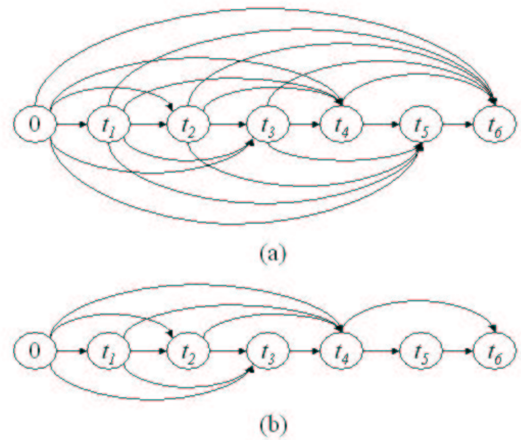

Figure 4. **(a)Composite tool sequence graph. (b) Forcing all solutions through critical tool t_4 splits the composite graph**

the graph $\{t_4, t_5, t_6\}$. 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, where where we have a sibling level consisting of features $\{f_1, f_2, f_3, f_4, f_5, f_6\}$ with critical tools $\{t_1, t_2, t_3, t_4, t_5, t_6\}$ respectively, the constrained graph will consist of the edges $\{e_{01}, e_{12}, e_{23}, e_{34}, e_{45}, e_{56}\}$ alone. The cheapest solution in this case will consist of the tool sequence $\{0 \rightarrow t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow t_5 \rightarrow t_6\}$. Clearly, this is inefficient as compared to the solution if the features were to be solved individually. This is particularly the case when each $f_i, i = 1, 2, 3, 4, 5, 6$ can admit all tools $t_j, j = 1, 2...i$ and has t_i as the critical tool.

5.2 Tweaking cheapest tool sequence solution

An alternative solution is to build the unconstrained graph (Figure 3 (a)), 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 $\{t_1, t_3, t_6\}$. Now we tweak this solution by adding an additional edge e^2_{34}(Figure 3 (b)). The weight of this edge is the cost of machining the decomposed feature D^2_{34} alone. This is done because, of all the edges $e^2_{i4}, i = 0, 1, 2, 3, 4$, e^2_{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. However we are not able to guarantee the optimality of this solution.

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 $\{t_1 \rightarrow t_2 \rightarrow t_4\}$ 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 the this solution is tweaked, we would add an additional edge e^2_{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.

5.3 Cascading tool sequence sub-graphs

Consider the example of two features f_1, f_2 in sub-section 6.1. We will start building the composite graph where every edges represents the total cost of machining decomposed features of both features by the tool named in the tail node of the edge. For example egde e_{ij} represents the cost of machining D^1_{ij}, D^2_{ij} by tool t_j. However, for edges $e_{(4-i)(4+j)}, \{i = 1,2,3,4\}, \{j = 1,2\}$, that span over the critical tool t_4 for f_2, 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 D^1_{35}, D^2_{35} by tool t_5. However, D^2_{35} does not exist. However, the node v_5 coceptually represents the state of the stock after all of f_4 has been machined. Therefore any edge $e_{(4-j)(4+j)}$ has to include the cost of completely machining f_4 after t_{4-j} is done machinig. The remaining part of this section will discuss the calculation of this cost.

Consider a sibiling level $S = \{f_1, f_2 \ldots . f_a, f_{a+1} \ldots . f_n\}$. Consider an edge $e_{(x-i)(x+j)}, \{i = 1,2,3,4..x\}, \{j = x+1, x+2 \ldots n\}$ that spans over a node v_x, representing a critical tool t_x for feature f_a. All tools $t_i, d_i \geq d_x$ can be used to machine all features in S . This is because for all such t_i, $l_i \geq h_k\{k = 1,2,\ldots a, a+1 \ldots n\}$. The head node of the edge, v_{x-i} represents the state of the stock after tool t_{x-i} is done machining in all features to the extent of its accessible areas. Therefore the shape of the stock after t_{x-i} is done machining is given by:

$$I_{x-i} = R - \sum X(A^{\beta}_{x-i}, h_{\beta}) - X(A^{a}_{x-i}, h_a), \beta = 1,2,\ldots a-1, a+1, \ldots .n \quad (7)$$

The node v_{x+j} conceptually represents the state after all of feature f_a whose critical tool is $t_x$is completely machined and features $\{f_1, f_2 \ldots . f_{a-1}, f_{a+1} \ldots f_n\}$ are machined to extent of accessible area of tool t_{x+j} in each of these. The shape of the stock after t_{x+j} is done machining is given by:

$$I_{x+j} = R - \sum X(A^{\beta}_{x+j}, h_{\beta}) - X(p_a, h_a), \beta = 1,2,\ldots a-1, a+1, \ldots .n \quad (8)$$

Note that n is the total number of features in the sibling level and $X(p_a, h_a)$ represents the removal volume of f_a.

The edge $e_{(x-i)(x+j)}$ represents the cost of machining $\{D^1_{(x-i)(x+j)}, D^2_{(x-i)(x+j)} \quad \ldots D^{a-1}_{(x-i)(x+j)}, D^{a+1}_{(x-i)(x+j)} \ldots D^n_{(x-i)(x+j)}\}$ in addition to the cheapest cost of machining whatever is left of f_a after tool t_{x-i} is done machining. This cheapest cost is obtaining by solving a separate graph $G_a(V = \{v_{x-i} \ldots v_x\}, e)$ starting with node v_{x-i} and ending with node v_x for f_a alone. The graph $G_a(V, e)$ is the cascading tool sequence sub-graph that needs to be solved for each edge that spans across the critical tool.

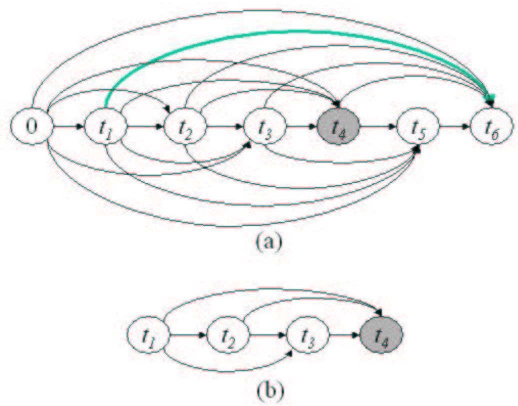

Figure 5. **(a)Edge e_{16} in composite tool sequence graph spans across critical tool t_4 for feature f_2. (b) Tool sequence sub-graph for feature f_2 associated with e_{16} of the composite graph.**

Figure 5 illustrates an example. There are two features, f_1, f_2 whose critical tools are t_6, t_4 respectively. The weight of the edge e_{16} in the composite graph is the cost of machining D^2_{16} in addition to the cost of the cheapest solution of the subgraph shown in Figure 5(b) for feature f_2 alone.

Once the composite graph has been built, it can be easily be solved for the shortest path to obtain the globally cheapest sequence for all the features in the sibling level.

6 Conclusions

In this paper tool sequence selection for a setup has been addressed. Two method were investigated. Once consisted of solving tool sequence graphs individually for each feature in the setup. This method does not consider air-time while calculating edge weights for the graph. When all the cheapest tool sequences have been found, tool paths used to machine each decomposed sub-features are connected on a per tool basis in a manner such as to minimize air time and preserve feature precedences.

The second method groups features into sibling levels depending on the various precedence constraints. Composite tool sequence graphs are built for each sibling level. Air-time for rapid traverse between decomposed sub-features in different features in the sibling level is accounted for, in the weights of the edges. Solving the tool composite tool sequence graph can lead to solutions that do not include critical tools for certain features. The concept of cascading tool sequence sub-graphs has been developed to solve this problem.

REFERENCES

[1] S. Gao and J. J. Shah. Automatic recognition of interacting machining features based on minimal condition subgraph. *Computer Aided Design*, 30(9):695–705, 1998.

[2] W. Regli. *Geometric Algorithms for Recognition of Features from Solid Models*. PhD thesis, University of Maryland at College Park, 1995.

[3] V. Sundararajan. *Feature Recognition and High-Level Process Planning for Rapid Prototyping Using Milling*. PhD thesis, University of California, Berkeley, 2000.

[4] J. Echave and J. J. Shah. Automatic setup and fixture planning for 3-axis milling. In *ASME Design Automation Conf., Las Vegas, NV*, 1999.

[5] B. Kannan and P. K. Wright. Efficient algorithms for automated process planning of 2.5-d machined parts considering fixturing constraints. *accepted for publication in the Intl. J. of Computer Integrated Manufacturing*, 2001.

[6] *MASTERCAM*. http://www.mastercam.com, 2002.

[7] *SURFCAM*. http://www.surfcam.com, 2002.

[8] M. Bala and T.C. Chang. Automatic cutter selection and cutter path generation for prismatic parts. *International Journal of Production Research*, 29(11):2163–2176, 1991.

[9] S. Arya, Cheng S.W., and D.M. Mount. Approximate algorithm for multiple-tool milling. *International Journal of Computational Geometry and Applications*, 11(3):339–372, 2001.

[10] Y. Chen, Y.S. Lee, and S.C. Fang. Optimal cutter selection and machining plane determination for process planning. *Journal of Manufacturing Systems*, 17(5):371–388, 1998.

[11] T. Lim, J. Corney, J.M. Ritchie, and D.E.R Clark. Optimizing automatic tool selection for 2.5d and 3d nc surface machining. *Computers in Industry*, 26(1):41–59, 2000.

[12] L. Kunwoo, T.J. Kim, and S.E. Hong. Generation of toolpath with selection of proper tools for rough cutting. *Computer Aided Design*, 26(11):163–180, 1994.

[13] J. Joo and H. Cho. Efficient sculptured pocket machining using feature extraction and conversion. *Journal of Manufacturing Systems*, 18(2):100–112, 1999.

[14] Y.S. Lee and T.C. Chang. Application of computational geometry in optimization of 2.5 d and 3d nc surface machining. *Computer in Industry*, 26(1):41–59, 1995.

[15] D. Veeramani and Y.S. Gau. Selection of an optimal set of cutting tool sizes for 2.5d pocket machining. *Computer Aided Design*, 29(12):869–877, 1997.

[16] Z. Yao, S. K. Gupta, and D. S. Nau. A geometric algorithm for finding the largest milling cutter. *SME Journal of Manufacturing Processes*, 3(1):1–16, 2001.

[17] R. D'Souza, P.K. Wright, and C.H. Séquin. Automatic microplanning for 2.5d pocket machining. *Journal of Manufacturing Systems*, 20(4):228–296, 2001.

[18] R. D'Souza, P.K. Wright, and C.H. Séquin. 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.