Technical Research Report

Selecting Flat End Mills for 2-1/2D Milling Operations

by Zhiyang Yao, Satyandra K. Gupta, Dana S. Nau

T.R. 2000-41

TECHNICAL RESEARCH REPORT

Selecting Flat End Mills for 2-1/2D Milling Operations

by Zhiyang Yao, Satyandra K. Gupta, Dana S. Nau

T.R. 2000-

SELECTING FLAT END MILLS FOR 2-1/2D MILLING OPERATIONS

Zhiyang Yao
Mechanical Engineering Department and Institute for Systems Research
University of Maryland
College Park,MD-20742
Email:yaodan@Glue.umd.edu

Satyandra K. Gupta
Mechanical Engineering Department and Institute for Systems Research
University of Maryland
College Park,MD-20742
Email:skgupta@eng.umd.edu

Dana S. Nau
Computer Science Department and Institute for Systems Research
University of Maryland
College Park,MD-20742
Email:nau@cs.umd.edu

ABSTRACT: The size of milling cutter significantly affects the machining time. Therefore, in order to perform milling operations efficiently, we need to select a set of milling cutters with optimal sizes. It is difficult for human process planners to select the optimal or near optimal set of milling cutters due to complex geometric interactions among tools size, part shapes, and tool trajectories. In this paper, we give a geometric algorithm to find the optimal cutters for 2-1/2D milling operations. We define the 2-1/2D milling operations as covering the target region without intersecting with the obstruction region. This definition allows us to handle the open edge problem. Based on this definition, we introduced the offsetting and inverse-offsetting algorithm to find the coverable area for a given cutter. Following that, we represent the cutter selection problem as shortest path problem and discuss the lower and upper bond of cutter sizes that are feasible for given parts. The Dijkstra's algorithm is used to solve the problem and thus a set of cutters is selected in order to achieve the optimum machining cost. We believe the selection of optimum cutter combination can not only save manufacturing time but also help automatic process planning.

1 INTRODUCTION

NC machining is being used to create increasingly complex shapes. Complex machined parts require several rouging and finishing passes. Traditionally, one single cutter is used in most cutting process. As the development of high speed tool change mechanism, tool change time is shorten to seconds. Consequently, multiple cutters are feasible to perform multi-pass cutting process for a given part. Selection of the right sets of tools and the right type of cutter trajectories is extremely important in ensuring high production rate and meeting the required quality level. It is difficult for human planners to select the optimal or near optimal machining strategies due to complex interactions among tools size, part shapes, and tool trajectories.

Many researchers have studied cutter selection problems for milling processes. There still exist significant problems to be solved in those approaches. Below are two examples:

- Most existing algorithms only work on 2-1/2*D* closed pockets (i.e., pockets that have no open edges), despite the fact that open edges are very important in general 2-1/2*D* milling.
- Most of the cutter selection approaches have their own limitations. No systematic definition and algorithm for cutter selection problem.

Finding the optimum cutters for a given part is supposed to help the process planning such as tool path planning and scheduling. Currently there is no such systematic approach that can do it automatically, therefore, we are going to given a systematical approach that can overcome the above limitations and can automatically generate the optimal cutter combination for 2-1/2 milling operations. We believe our research result can also be integrated into some commercial CAM system such that the cutter selection, CNC code generation and process planning can be automatically performed.

2 RELATED WORK

More and more people have realized that finding the optimal cutter combination is very helpful in the process planning. There are several papers talk about cutter selection problem for 2-1/2D milling process.

D.C.Yang and Z.Han present the tool selection problem in 3-axis NC machining for free-form surfaces [Yang 99]. In their paper, they assume that during milling process, only limited tools are available; and tool change time is a constant number. They think too many tool changes is generally not desired in machining practice as it can affect surface finish and cause unnecessary tool ware. In their procedure, the total machining time is the sum of cool cutting time and tool change time. By enumerate all possible combinations of given set of cutter, they can find the combination with minimal machining time. Their approach limits the number of tool changes to within 3, which restricts the cutter combination. Yang et.al. find the approximate length of the cutter path, and used the tool path length and the feed-rate to estimate the machining time. By using this method, they assume the zig-zag tool path is used such that they can estimate the tool path length by the machining area and the cutter radius.

Sarma et.al. discussed the cutter selection in 3-axis rough cutting process [Maha 97]. They used Voronoi diagram to find the offsetting area, which only handle the closed pocket problem. In their approach, a table in which the area removed and the cutter size can be stored is build by using offsetting/ reverse-offsetting idea to get the cuttable area for a cutter. They also assume the total machining time is the sum of cutting time and tool changing time and the tool cutting time is proportional to the division of area to be cut and the cutter's diameter. Instead list all possible combinations, they check the order of cutters in those combination and then choose the combination, which from large tool to small too and then get the machining time for this combination. And finally they find the best combination.

D.Mount et.al presents an approximation algorithms for finding the optimized multiple tools for milling process [Arya 98]. They transform the milling problem to a weighted set-cover problem using a greedy strategy to obtain a logarithmic ratio. Their algorithm works only for pocket with connected domain. In their cost model which is used to find the optimum tool set, the cutting time, the tool changing time and the tool transporting time are considered.

Lee et.al., assume two tools are used in rough cutting, bigger one used for the portion with a simple shape while a small tool should be used for the complex portion [Lee 94]. They used the Octess method to find the approximately select the cutters. This method is only valid for rough cutting, also they don't have a systematically theory of why the two cutter are optimum.

T.C.Chang et.al., presented a method to find the cutter set for prismatic parts [Bala 91]. Their basic idea is trying to fit the possible large circle into contours to select possible large cutter to save processing time. They take both cutter change time and geometric constraints into consideration. They stated the problem as follows. If there exists a set of cutters, and after machining with these cutters, only finishing machining is needed for fillet radii corners, so the problem is to determine the cutter with the largest radius in this set.

Veeramani and Gau used dynamic method in cutter selection problem [Veer 97]. Their algorithm only works for closed pocked. Also they assume that the smallest cutter should be the minimal radio of the corners. They use two stages to do the work: first using Voronoi mountain to get the relationship between the machinable area and the cutter size, then they use dynamic programming to optimize the result. The goal of using dynamic programming of identify an optimized set of cutting tools is to find the route from stage 0 to stage N such that the objective function is optimized. Only tool changing time and cutting time are considered in their approach. The total processing time for a cutting-tool includes the time spent by the cutting-tool in material removal movements as well as non-material-removal movements. They use the Voronoi Mountain to estimate the area that can be cut by a cutter and by using the offset idea, the approximate cutter path length by using contour - parallel tool path can be estimated such that the total cutting time can be estimated too.

In summary, most existing cutter selection algorithms select milling cutters by minimizing the machining time. Moreover, most previous algorithms have the following restrictions: (1) they can only handle pockets with closed edges, (2) most approaches restrict the number of cutters that can be selected.

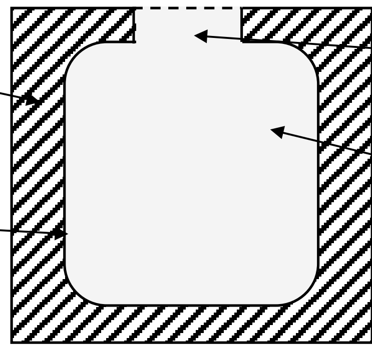


Figure 1: Problem Formulation

Our paper describes a systematic algorithm for finding an optimal set of milling cutters for 2-1/2D milling operations. In selecting milling cutters we consider both the tool changing time as well as machining time and generate solutions that allow us to minimize the total manufacturing time. Our tool selection algorithm improves upon the previous work in the following manner: (1) it can handle both closed as well as open edges, (2) in selecting cutters it does not restrict the number of cutters.

Currently our algorithm is restricted to 2-1/2D milling operations. In Section 3, we define the 2-1/2D milling operations as the problem of covering a target region with a cylindrical cutter without intersecting with the obstruction region. This general definition allows us to handle both open and closed edges. In Section 4, we introduce the method to compute the coverable area for each given cutter. In general, smaller the cutter size, the more area it can cover. We find the coverable area by offsetting the obstruction region. After computing coverable area for each cutter, in Section 5, the shortest path problem is used to represent the problem if finding a sequence of cutters for one part. We will mainly we will discuss what the lower and upper bond of feasible cutters for a given problem according to geometric and machining constraints. Then the Dijktra's algorithm is used to solve the problem. Several examples will be discussed in this section. The conclusion and discussion of our work will be given at the final section of this paper.

3 PROBLEM FORMULATION

The most common milling problem is the problem of cutting a given 2-*D* region at some constant depth using one or more milling tools. We define the region to be machined as the *target region T*. Beside the target region, there is also an *obstruction region O* which is the region that the cutting tool should not cut during machining [Yao00a, Yao00b]. An example is shown in Figure 1. The target region and the obstruction region must both be regular sets, but may each consist of a number of non-adjacent sub-regions:

$$T = T_1 \cup \ldots \cup T_j\,;$$

$$O = O_1 \cup \ldots \cup O_k.$$

In our research, we assume that the boundary of each sub-region consists of only of line segments and segments of circles.

The *target boundary* B_T is the boundary of the target region, and the *obstruction boundary* B_O is the boundary of the obstruction region. The edges on the obstruction boundary are called *obstruction edges*. We call an edge of the target boundary a *closed edge* if it is coexists with an obstruction edge; otherwise we call it an *open edge*. Figure 2 shows examples of open and closed edges. We will use the dashed line to represent the open edges, the shaded regions as the obstruction region.

Let C be a circular cutter of radius r, and (x, y) be a point. Then the *region covered* by C at the point (x, y) is the following set:

$$R(x,y) = \left\{ \text{ all points } (u,v) \text{ such that } \sqrt{(u-x)^2 + (v-y)^2} \le r \right\}$$

A point (x, y) is a *permissible location* for C if the interior of $R(x, y)$ does not intersect with the obstruction region, or equivalently, if $O \cap^* R(x,y) = \varnothing$. A set of points S is *coverable* using a cutting tool C if for every point p in S, there is a permissible location of C that covers p.

We are using the *region covering* idea in our following research. That is, we are trying to find the optimal cutters that can cover the target region without intersecting the obstruction region by optimal total machining time. By using this definition, we can not only handle the problem with closed edge, but also the problem with open edges.

Based on the region covering idea, given a cutter, we call the area of the target region that can be covered by this cutter as *coverable area of this cutter* $A(C)$. Knowing the coverable area for a given cutter, we can estimate the cutting time if we use the cutter in machining process.

In our research, the total time of machining a part from loading the stock to the finish cutting is called the *total machining time* T_M. The purpose of our research is to find the best combination of cutters such that the total machining time for a given part is minimal. In the milling operations, the total machining time is composed by the following items:

1. Total real cutting time T_{ct} (the time spend on moving cutters to cut the profile);
2. Total cutter change time T_{cc} (the total time of changing tools that are already loaded in the tool magazine during machining operations);

Mathematically: $T_M = T_{ct}+T_{cc}$.

Our research is focusing on the 2-1/2D milling problem. We mainly consider the geometric information in finding the cutters. Here are some assumptions:

1. In the milling operations, if we are going to use multiple cutters for one part, we will use the cutters in the sequence of decreasing size.
2. A smaller cutter is only used to cut the area left by bigger ones.
3. The tool change time can be estimated by the average experimental operations time.

Based on our definition and assumption, we define the *cutter selection problem* as following. Given one part involves several 2-1/2D milling operations, and a set of cutting tools, we are going to find the best combination of milling tools such that the total machining time including tool changing time and cutting time for the part is minimal.

Mathematically speaking, we have cutter set $C=\{C_1,C_2,\ldots,C_n\}$ where the cutters are sequenced in a decreasing order such that the cutter radius have the relationship $r_1>r_2>\ldots>r_n$, and a part P, we need to find a subset of cutters $C^*\subseteq C$, $C^*=\{C_1^*, C_2^*,\ldots,C_m^*\}$ where $m\leq n$ and the cutters in C^* are also in decreasing sequence, to achieve the minimal total machining time T_M .

It is difficult to select an optimal set of cutters for complex parts by hand. Therefor, in this paper, we are going to

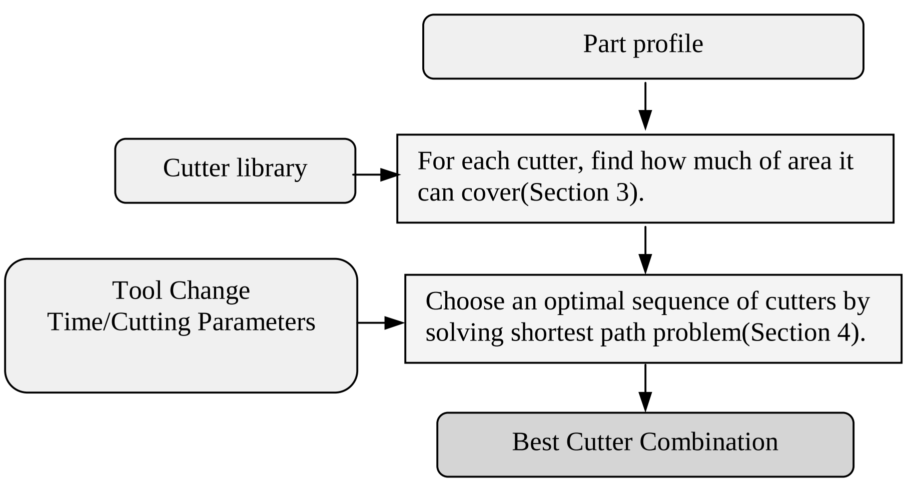


Figure 2: Overview of Our Approach

give a set of algorithms such that the cutter selection problem can be solved automatically. As shown in Figure 2, based on this definition, by given a part, we can get the profile of target and obstruction regions of is. Suppose we are given a set of cutters, we can find the coverable area for each given cutter using our algorithm. Then a shortest path algorithms is used in order to find the best sequence of cutters that can cut all the parts by minimal total machining time including cutting time and tool changing time.

4 FINDING COVERABLE AREA FOR A GIVEN CUTTER

In cutter selection problems, multiple cutting passes are used. The bigger cutter is usually used to cut material as fast as possible and then smaller cutters are used until the whole target region is covered. How to find the feasible cutter size and determine the coverable area for each cutter for a given part is believed to help the cutter selection problem. In this section, we will introduce several geometric algorithms to automatically calculating the coverable area for a given part.

4.1 PROFILE EXTRACTION ALGORITHM

Before we go further about the problem of finding multiple cutters for multiple part, we will first introduce how to get the target and obstruction region profile from the CAD model. The idea of extracting the profile form a 3-*D* model is following. Suppose we have a 3-*D* model of a given part, and we know the cutting depth for the cutting process, we can use a surface to intersect with the 3-*D* model in different height, i.e., the upper surface and bottom of the cutting stage, and then we can check the difference of those two intersection profile and find the real target and obstruction profile. Figure 3 shows an example of how this procedure works.

Suppose we are given a 3-D model of the final part P, we know before the cutting happens, the upper surface is at height h_2 and cutter should cut the stock at the bottom surface at height h_1, so we can use the following algorithm to get the obstruction and target region.

PROFILE_EXTRACTION(P, h_1, h_2)

1. Get the intersection surface at height h_1 as S_1
2. Get the intersection surface at height h_2 as S_2
3. Obstruction region = S_2, while target region = $S_1 -^{*} S_2$

4.2 COVERABLE AREA FINDING ALGORITHM

Intuitively, the *offset region* $F_j(C)$ is the region formed by offsetting the obstruction subregion O_j using the radius r of the cutter C. Mathematically speaking, a point p is in $F_j(C)$, if $p \notin O_j$ and the minimal distance from p to any point in O_j is less or equal to r. The regular union of all $F_j(C)$*s* is called the offset region $F(C)$ for all obstruction regions. Figure 4(b) shows an example of an offset region.

After we get the offset region for all obstructions, we can define the *inverse-offset region* $I(C)$ of $F(C)$ as the region formed by inverse offsetting the outer boundary of the offset region by using the radius r of the cutter C. Mathematically speaking, a point p is in $I(C)$, if $p \in F$ and the minimal distance from p to any point on the outer boundary of F is less or equal to r. Figure 4(c) shows an example of an offset region. From the definition we know that $I \subseteq F$.

There may be difference between the offset region F and the inverse-offset region I of F, we define the uncoverable region as the region $U=T\cap^{*}(F-^{*}I)$. Figure 4(d) shows an example of an uncoverable region.

From the definitions, we can get the following lemma:

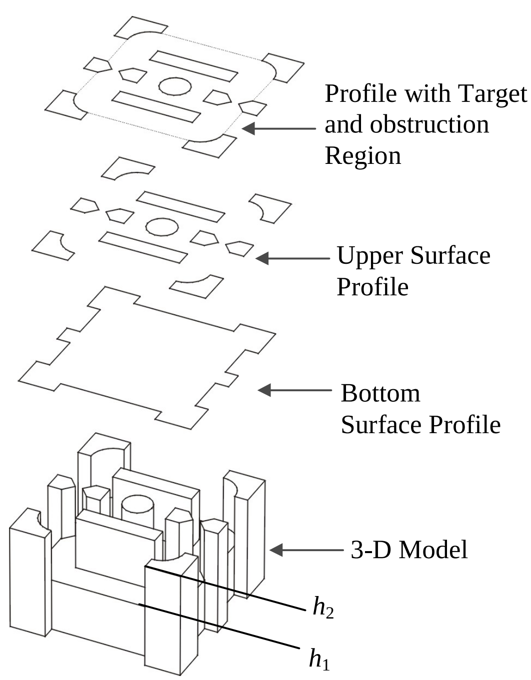


Figure 3: Example of profile extraction

Lemma 1: We can not locate the cutter *C* centered at any point inside of *F* (not including the outer boundary of *F*) without intersecting with *O*.

Proof: Suppose we locate cutter *C* at point *p*, $p \in F$, we know that the minimal distance from *p* to any point in O_j is less or equal to *r* (only if the point is on the boundary of F can the equality happens), so the distance from the center to at least one point of C will greater than *r*, so *C* must intersect with part of *O*.

Lemma 2: For a given cutter *C* and the part with obstruction *O* and target region *T*, the points in the uncoverable region cannot be covered by the cutter *C*.

Proof: For a given point *p*, $p \in U$, if we can put a cutter *C* that cover p, then, the center of *C* must locate inside of *F*, from Lemma 1, we know that if we locate *C* at any center point inside *F*, *C* will intersect with *O*, that means *p* is not coverable by *C*.

Lemma 3: Any point in the target region beside the uncoverable region is coverable.

Proof: From the definition, we know that all points outside of *U* in *T* should whether locate outside of *F*, or should

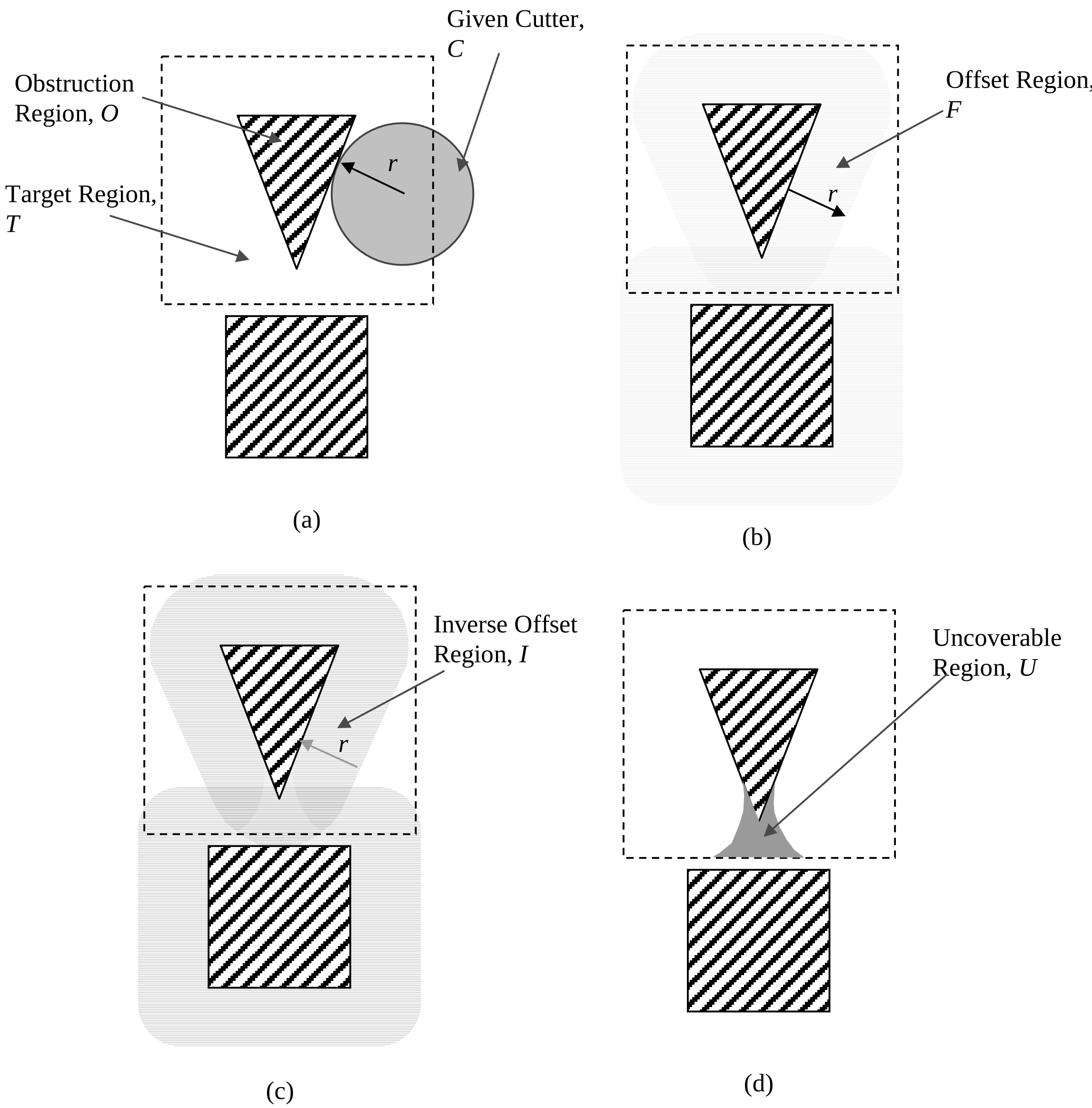


Figure 4: Finding Coverable Area by Offsetting and Inverse-offsetting

belong to I. For the points outside of F, we know that if we locate a cutter at that point, the distance from its center to any point in O is greater than r, so this point is coverable. For any point inside of I, we can locate a cutter centered at the point on the boundary of F that has the shortest distance from p, we know that the cutter will not intersect with O, so the point is coverable.

By knowing the properties of the offset and inverse offset region, we can use the offsetting and inverse offsetting idea to find the coverable area for a given cutter. For a given part, we know the obstruction regions and the target regions. We can first offset each obstruction by the given radius r of a given cutter. Then we do union operation of all the offset regions if they are contacted with another offset region. After that we do inverse offsetting based on those union regions, there maybe some small part of the offset region that is outside of the inverse offset region, that is the uncoverable region. Finally, the area of the coverable region is the difference of the target region and the uncoverable region. Figure 4 shows how the procedure works.

Suppose the obstruction region of the given part is composed by several disconnected regions $O=\{O_1,O_2,\ldots O_n\}$; the target region of the part is represented by T, for a cutter with radius r, the algorithm to find he coverable area is show a s following:

COVERABLE_AREA_FINDLING (O, T, r)

1. for each disconnected obstruction region O_i, do
2. offsetting by r, the new region is called offset region F_i
3. if F_i intersects some other offset region, unite those connected regions together
4. for each F_j, do
5. inverse offsetting F_j by r and the region called I_j
6. find $U_j=T\cap^*(F_j-^*I_j)$
7. $U= U \cup^* U_j$
8. Return area of A = area of ($T-^*U$)

Correctness proof of COVERABLE_AREA_FINDLING: From Lemma 2 and Lemma 3, we know that the coverable area for a given cutter is the area of the target region minus the uncoverable region. Out algorithm is guaranteed to find the right coverable area for a given cutter. Meanwhile, if we are given a part and a cutter, the algorithm can find the offset and inverse-offset region, such that the answer can be found.

5 CUTTER SELECTION USING SHORTEST PATH ALGORITHM

By given a 2-1/2D milling problem, if we know the size of available cutters, we can use the algorithm in section 4 to find the coverable area for each cutter. After we get the relationship of the coverable areea and the cutter size, we are ready to find the optimal sequence of cutters such that by using those cutters, the total time of machining given part is minimal. We cast the cutter selection problem into the shortest path problem and sue the Djikstra's algorithm to solve it.

5.1 BACKGROUND ON SHORTEST PATH PROBLEM AND DIJKSTRA'S ALGORITHM

In a *shortest-paths problem*, we are given a weighted, directed graph $G = (V, E)$, with weight function w: $E\rightarrow\mathbf{R}$ mapping edges to real-valued weights (as shown in Figure 5). The weight of path $p = \langle v_0, v_1,\ldots,v_k\rangle$ is the sum of the weight of its constituent edges:

$$w(p)=\sum_{i=1}^{k} w(v_{i-1},v_i).$$

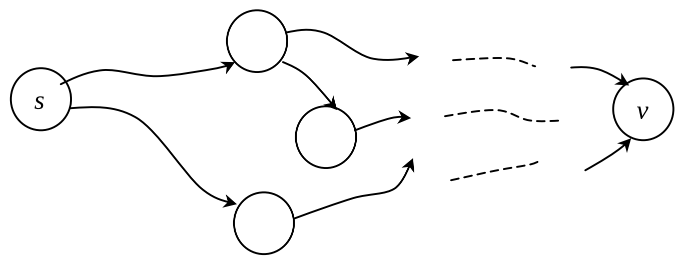


Figure 5: Shortest path Problem

We define the shortest-path weight from u to v by

$$\delta(u,v) = \begin{cases} \min\{w(p): u \xrightarrow{p} v\} & \text{if there is a path from } u \text{ to } v \\ \infty & \text{otherwise} \end{cases}$$

A *shortest path* from vertex u to vertex v is then defined as any path p with weight $w(p)=\delta(u,v)$.

The *single-source shortest-paths problem* is defined as: by given a graph $G = (V, E)$, we want to find a shortest path from a given source vertex $s \in V$ to every vertex $v \in V$.

Dijkstra's algorithm is known to be a good algorithm to solve the single-source shortest-paths problem on a weighted, directed graph $G = (V, E)$ for the case in which all edge weights are nonnegative. Therefor, in G, we assume that $w(u, v)\geq 0$ for each edge $(u, v)\in E$. In order to indicate the vertices on the shortest paths, given a graph $G=(V,E)$, we maintain for each vertex $v\in V$ a predecessor $\pi[v]$ that is either another vertex or NIL.

The following theorem shows the correctness of Dijkstra's algorithm: If we run Dijkstra's algorithm on a weighted, directed graph $G=(V,E)$ with nonnegative weight function w and source s, then at termination, $d[u] = \delta(s, u)$ for all vertices $u\in V$. The running time required by Dijkstra's algorithm is $O(V^2)$ [Thom 1990].

5.2 CUTTER SELECTION PROBLEM REPRESENTATION

Assume the cutters $\{C_1,C_2,\ldots,C_n\}$ are given in order of decreasing size. Initially, we have the stock for given part, after use the selected cutters, the initial stock will be machined into the final part. Here, we define the machining process as several machining stages; each stage represents the intermediate part shapes, which are gotten by using the corresponding cutter on the initial stock. P_0 is the initial stage in which no cutter is used. P_n is the stop node in the graph. P_n represents the last stage in which a cutter C_n is used such that all target area left after using bigger cutters should be cleaned. In other word, C_n should be a cutter that can cover all target regions. We will build a graph in which each node represents a machining stage. The graph is shown in Figure 6. The *i*'th node represents the part shape P_i after the *i*'th cutter has been used on the stock. Each edge represents the operation of using some cutter on the part. In this graph, the cost of each edge (i, j) is the time (cutting time and tool change time) needed to use cutter C_j on parts P_i.

By using this representation, the optimal manufacturing time is equal to the least cost of any path from P_0 to P_n. After we get this graph, we can easily solve this problem by using Dijkstra's algorithm.

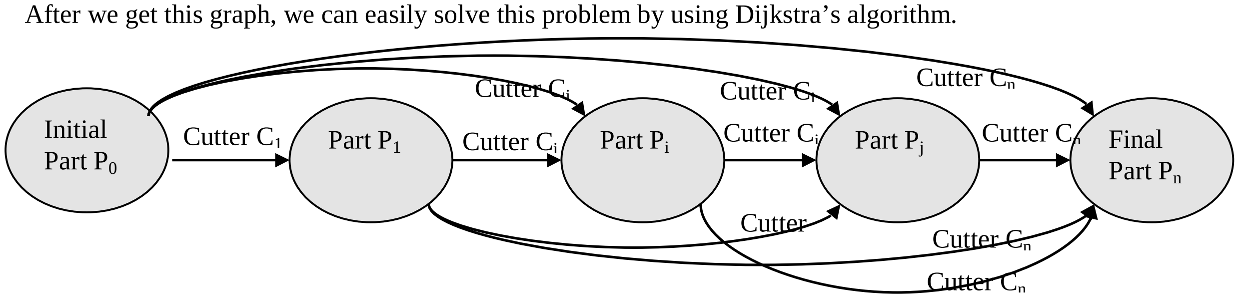


Figure 6: Problem Representation

5.3 USING DIJKSTRA'S ALGORITHM IN FINDING OPTIMUM CUTTER SET

In order to calculate the weight for each edge, i.e., the machining time, we have the following assumptions:

1. T_{cc} is known as average cutter change time in terms of k_c which can be determined by experimental operations.
2. During machining, for a cutter C whose diameter is d, if the area really cut by only use C is A', then the total cutting time can be estimated by the formula: $T_{ct}=k\times A'/r$
3. If a cutter is used during machining operations, the cutter will cut all possible area left by the formal cutters. That means, if the current cutter is c_j and it is used after cutter c_i, then the real area cut by cutter c_j should be A_j-A_i.
4. During machining operation, we assume the cutters are used by a decreasing order, i.e., the larger tool is always used before a smaller tool can be used. The assumption is made based on the observation that usually for a given area A', the bigger cutter used to cut it, the less time spend on machining. Following this assumption, we can calculate the total cutting time for a given cutter C_j if it is used right after C_i is used, where $r_j<r_i$: $T_{cti} = k\times(A_j-A_i)/r_j$, k is a given ratio controls the cutting time for different machines. k can also be obtained by experimental operations.
5. Cutter C_n is the lower bond of all feasible cutters. As we need to clean the whole target region, C_n must be the end node of the shortest path.

The CUTTER_SELECTION procedure (**CS**) works this way: first, we will build a weighted, directed graph $G=(V,E)$ with nonnegative weight function w and source node s and target node v. We call this procedure as BUILD_GRAPH. In this graph, each node represents a cutter size in the sequence from bigger to smaller. The first node (the source node s) represents a virtual cutter c0 whose diameter is infinity and whose coverable area is 0. The last node represents the cutter found in the given cutter list whose coverable area should be the of the whole target region. The edge links cutter i and j (here, the size of cutter i is bigger than cutter j) in the graph edge $e(i,j)$ represents one possible cutting sequence, i.e., after using cutter i, we changed into using j to cut the left area. The weight w of the edge $e(i,j)$ is defined as:

$$w(i,j)= T_{cc}+T_{ctj}=k_c+ k\times(A_j-A_i)/r_j$$

After we get graph G, the optimization problem becomes fining the shortest path begins from the cutter C_0 whose diameter is infinity to the cutter C_n whose diameter if the smallest one among all cutters. We can use the Dijkstra's algorithm to find the shortest path, by means, the best cutter sets in decreasing order that can machine the given part in minimum total machining time.

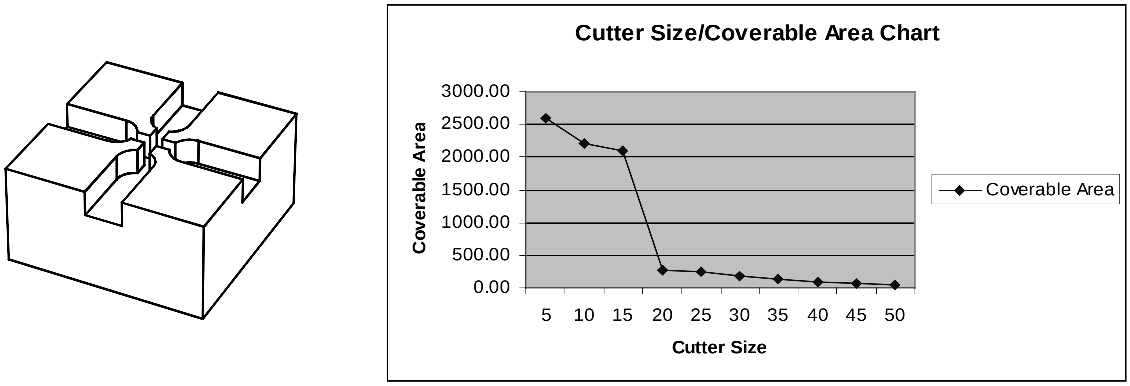
CUTTER_SELECTION(r[i],A[i], i=1 to n)

1. BUILD_GRAPH
2. DIJKASTRA(G, w, $s=0$, $v=n$)
3. PRINT_RESULT

6. ALGORITHM ANALYSIS

Suppose we have N cutters. For the given part, the number of the edges of the obstruction region is E. The complexity of our algorithm can be gotten by:

1. For each offsetting operations, we know the average complexity is O(E) [Yang 93]. Meanwhile, the inverse-offsetting operation is based on the offset region, the segment should be O(E) so that the complexity of doing inverser-offsetting is also O(E). Thus, the total time for finding coverable area for each part corresponding to all N cutters is O($N\times E$).



(a) Given Part (b) Chart

Figure 7: Cutter Size/Coverable Area Chart for a Given Part

2. The Dijkstra algorithm runs in $O(V^2)$, V is the vertex in the graph. So for the cutter selection problem, the running time in shortest path selection phase is $O(N^2)$.

In summary, the total running for our algorithm is $O(N\times E+ N^2)$.

7. IMPLEMENTATION AND EXAMPLES

We have implemented our algorithm by using C++ and ACIS. Here shows one example. Suppose for part shown in Figure 7(a), we are initially give 10 cutters, ∅5, ∅10, ∅15, ∅20, ∅25, ∅30, ∅35, ∅40, ∅45, ∅50. By using our algorithm, we can build a cutter size and coverable area relationship char as shown in Figure 7(b).

By using the shortest path problem, we can get some results as discussed following: if we assign different tool changing time and cutting factors, we are going to have different cutter combinations as shown in Figure 8. In Figure 8, we can see that if the cutter changing time plays a more and more dominate role in total machining time, the total number of cutters in the optimal set will be less.

8. DISCUSSION AND CONCLUSION

Competition in manufacturing market requires optimal machining time and cost. How to automatically select an optimal set of cutters such that the total machining time is minimal is very important. As it is difficult to do this cutter selection work manually, we give a series of algorithms such that we can select one or multiple optimal cutters for a given part. In particular, the contributions of our research are:

- We define the region covering idea in cuter selection problem. By this definition, we cannot only handle the problem with closed edge but also the problem with open edges. By this definition, not only the best cutter sequences can be selected, but also can help us determine the efficient cutter path.
- The method of how to extract the profile in name of target and obstruction region from a given 3-D model is given. By using this profile extraction algorithm, we can ensure this whole system can be performed automatically.
- We discussed the upper and lower bond of a set of cutters for a given part, and give the offset and inverse offset algorithm to find the coverable are for a given cutter.
- We represent the cutter selection problem as shortest-path problem, and by solving this problem the optimal

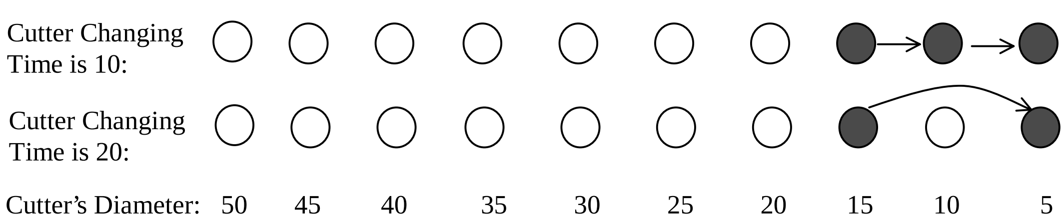


Figure 8: For different cutting parameter, the result is different

sequence of cutters for multi-pass milling can be found.

We plan to extend our work in the following areas to overcome current limitations:

- Currently when we estimate the real cutting time, we assume the cutting time proportional to the division of coverable area and the cutter size. In practice, the cutting time will also depend on the cutter path. We will try to consider tool path information such that the result will be more suitable.
- Tool life will also constrain tool selection. We should avoid the situation that the tool is broken during machining process. We will consider tool wear rate and total cutting area in order to use tool life information in cutter selection problem.
- So far, we are mainly consider the geometric constraint in cutter selection problem, we will exploit more about machine constraints

ACKNOWLEDGMENTS

This research has been supported by the NSF grants DMI9896255 and DMI9713718. Opinions expressed in this paper are those of authors and do not necessarily reflect opinion of the National Science Foundation.

REFERENCE

[Arya 98] S. Arya, S. W. Cheng and D. M. Mount. Approximation algorithm for multiple-tool milling. Proc. Of the 14th Annual ACM Symposium on Computational Geometry, 1998, pp. 297-306.

[Bala 91] Bala, M and Chang, T C, Automatic cutter selection and optimal cutter-path generation for prismatic parts. International Journal of Production Research, 1991, 29(11), 2163-2176.

[Lee 93] Y. S. Lee, B. K. Choi and T. C. Chang. Cut distribution and cutter selection for sculptured surface cavity machining. International Journal of Production Research, 1993, 30(6), 1447-1470.

[Lee 94] K. Lee, T. J. Kim and S. E. Hong. Generation of toolpath with selection of proper tools for rough cutting process. Computer-Aided Design, vol(26) 822-831, Nov. 1994.

[Lee 95] Y. S. Lee and T. C. Chang. Application of computational geometry in optimization 2.5D and 3D NC surface machining. Computers in Industry, 1995, 26(1), 41-59.

[Lee 96] Y. S. Lee and T. C. Chang. Automatic cutter selection for 5-axis sculptured surface machining. International Journal of Production Research, 1996, 34(4), 977-998.

[Li 94] H. Li, Z. Dong and G. W. Vicker. Optimal toolpath pattern identification for single island, sculptured part rough machining using fuzzy pattern analysis. Computer Aided Design, 1994, 26(11), 787-795.

[Maha 97] B. Mahadevan, L. Putta and S. Sarma. A feature free approach to tool selection and path planning in 3-axis rough cutting. Proceedings of First International Conference on Responsive Manufacturing, Nottingham, September 1997, pp.47-60.

[Thom 90] Thomas .H. Cormen, C.E. Leiserson, and R.L. Rivest, Analysis of Algorithms, The MIT Press/McGraw Hill, 1990.

[Veer 97] D. Veeramani, and, Y. S. Gau. Selection of an optimal set of cutting-tool sizes for 2.5D pocket machining. Computer-Aided Design, 1997, 29(12), 869-877.

[Yang 99] D.C.H. Yang and Z. Han. Interference detection and optimal tool selection in 3-axis NC machining of free-form surface. Computer-Aided Design, 1999, Vol.31, pp.303-315.

[Yang 93] Shi-Nine Yang and Ming-Liang Huang, A new offsetting algorithm based on tracing technique, 2nd ACM Solid Modeling, 93.5, Montreal, Canada, pp.201-210

[Yao00a] Z. Yao, S.K. Gupta, and D. Nau. A geometric algorithm for finding the maximal cutter for 2-D milling operations. In *International Conference on Flexible Automation and Intelligent Manufacturing*. College Park, MD, June 26-28, 2000.

[Yao00b] Z. Yao, S.K. Gupta, and D. Nau. Finding the maximal cutter for 2-D milling operations. In *Proc. DETC 2000: 2000 ASME Design Engineering Technical Conference*. Baltimore, MD, September 10-13, 2000