CONCLUSION

Machine-tool chatter remains a significant factor influencing the surface finish and the increased in cost for machining. If our goal as researchers is to provide manufacturers effective modelling tools that create robust productivity and drive performance to new levels of customer demands, the prediction and elimination of chatter require the examination of both the deterministic and stochastic stability of the machining process. Efforts in the literature are so far mostly devoted to deterministic analyses. In this paper sufficient conditions for stable machining subjected to nonlinear deterministic and linear stochastic restoring force have been derived. Explicit expressions for the onset of chatter at a Hopf bifurcation and the largest Lyapunov exponent during orthogonal machining operation are obtained.

REFERENCES

1. **Tobias, S. A.**, *Machine Tool Vibration*, Blackie and Sons Ltd, London, Glasgow, 1965.

2. **Hanna, N. H. and Tobias, S. A.**, A theory of nonlinear regenerative chatter, ASME Journal of Engineering for Industry, 96, 247-255, 1974.

3. **Tlusty, J. and Ismail, F.**, Basic nonlinearity in machining chatter, Annals of the CIRP, 30(1) 1060-1067, 1981.

4. **Kondo, Y., Kawano, O. and Sato, H.**, Behaviour of self-excited chatter due to multiple regenerative effect, ASME Journal of Engineering for Industry, 103, 325-329, 1981.

5. **Yoshitake, Y., Inoue, J. and Sueoka, A.**, Vibrations of a forced self-excited system with time lag, *Bulletin of JSME,* 26 (221) 1943-1951, 1983.

6. **Shi, H. M. and Tobias, S. A.**, Theory of finite amplitude machine instability, *International Journal of Machine Tool Design and Research*, 24 (1), 45-69, 1984.

7. **Minis, I. E., Magrab, E. B. and Pandelidis, I. O.**, Improved method for the prediction of chatter in turning, Part I: determination of the structural response parameters, ASME Journal of Engineering for Industry, 112, 12-20, 1990.

8. **Stépán, G. and Nagy, T. K.**, Nonlinear regenerative machine tool vibrations, *DETC97/VIB-4021, ASME Design Engineering Technical Conference*, September 14-17, Sacramento, California, 1-11, 1997.

9. **Stratonovich, R. L.**, *Topics in the Theory of Random Noise II,* Gordon and Breach, New York, London, Paris, 1967.

10. **Mao, X.**, *Exponential Stability of Stochastic Differential Equations,* Marcel Dekker Inc., New York, Basel, Hong Kong, 1994.

11. **Hale, J. K.** *Theory of Functional Differential Equations*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1977.

12. **Hale, J. K. and Lunel, S. M. V.**, *Introduction to Functional Differential Equations*, Springer-Verlag, New York, Berlin, Heidelberg, London, Hong Kong, Budapest, 1993.

13. **Hale, J. K. and Perelló, C.**, The neighbourhood of a singular point of functional differential equations, *Contributions to Differential Equations*, 3, 351-375, 1964.

A GEOMETRIC ALGORITHM FOR FINDING THE MAXIMAL CUTTER FOR 2-*D* MILLING OPERATIONS

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

University of Maryland
College Park

ABSTRACT: In this paper, we present a geometric algorithm of finding the maximal cutter for 2-*D* milling operations. Our algorithm works not only for the common closed pocket problem, but also for the general 2-*D* milling problems with open edges. We define the general 2-*D* milling process in terms of a "target region" to be machined and an "obstruction region" that should not intersect with the cutter during machining. Our algorithm finds the largest cutter that can cover the target region without interfering with the obstruction region. Finding the biggest cutter is expected to help in the selection of the right sets of tools and the right type of cutter trajectories, and thereby ensure high production rate and meet the required quality level.

INTRODUCTION

Selecting the right sets of tools and the right type of cutter trajectories is extremely important in order to ensure high production rate and meet required quality levels. For this reason, how to select an optimal cutter (or an optimized set of cutters) is an active topic of research in the process planning area. [1-6] Finding the maximal cutter is one of the most significant steps in order to optimize the cutters. Although many researchers have studied maximal-cutter-finding problems for milling processes, there still exist significant problems to be solved. Below are two examples:

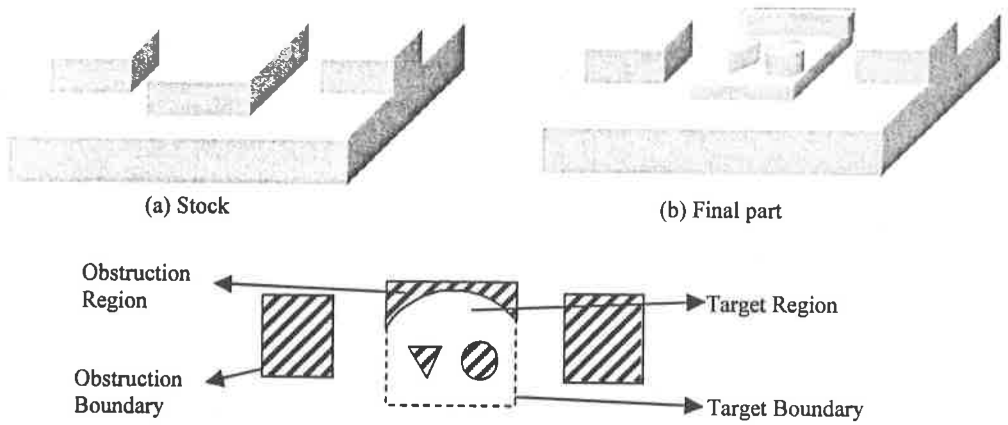
- Most existing algorithms only work on 2-*D* closed pockets (i.e., pockets that have no open edges), despite the fact that open edges are very important in general 2-*D* milling.
- Since there are several different definitions of what it means for a cutter to be feasible for a region, different algorithms that purport to find the largest cutter may in fact find cutters of different sizes.

In this paper, we present an algorithm for finding the maximal cutter size for general 2-*D* milling process. This algorithm overcomes the above-described problems.

PROBLEM FORMULATION

Finding Maximal Cutter for General 2-*D* Milling Process

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. In addition to the region to be machined (which we call the *target region P*), there is also an *obstruction region O* (regions which should not interfere with tool during machining). An example is shown in Figure 1. Neither the targer region nor the obstruction region needs to be completely connected; in general, they can consist of several unconnected sub-regions. The boundary of each sub-region is composed by connected edges; and the edges we consider in this paper are either line segments or segments of circles.



(a) Stock (b) Final part

(c) Target and Obstruction Regions

Figure 1: Notation and Nomenclature

We classify the boundaries of the two regions into two different kinds of boundaries: a *target boundary* B_P (the boundary of the target region), and an *obstruction boundary* B_O (the boundary of the obstruction region). Figure 1(c) shows an example of the target boundary and obstruction boundary. The edges on the obstruction boundary are called *obstruction edges*. We call an edge of the target boundary a *closed edge* if it is also an obstruction edge; otherwise we call it an *open edge*. (Note that 2-*D* closed pockets do not have any open edges. [1, 3-4]) Each closed edge or obstruction edge separates the *material* (i.e. part of the obstruction region) from part of the target region. The side on which the material lies is called the *material side* and the other side is called the *non-material side*. We will use the dashed line to represent the open edges, the shaded regions as the obstruction regions.

Preliminary Definitions

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

$$R(x,y) = \{ \text{ all points } (u,v) \text{ such that } \sqrt{(u-x)^2+(v-y)^2} \le r \ \} \quad (1)$$

   An example is shown in Figure 2.

2. *Permissible Locations*: A point *(x,y)* is a *permissible location* for *T* if the interior of *R(x,y)* does not intersect with the obstruction region. An example is shown in Figure 2. This can be mathematically expressed as: $O \cap^* R(x,y) = \phi$ (2)

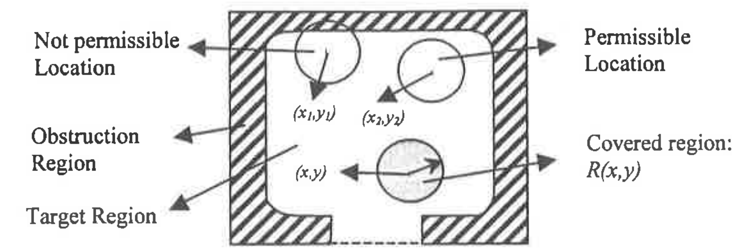
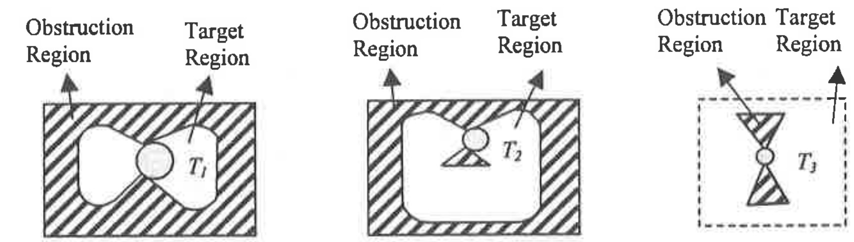


Figure 2: A covered region, and locations that are permissible and not permissible.



(a) Based on Alternative 1 (b) Based on Alternative 2 (c) Based on Alternative 3

Figure 3:Three Maximal Cutters Found by Different Alternatives of Cutting Feasibility

3. *Cutter Feasibility for a Target Region*: Most existing algorithms for cutter-tool-size selection are just used to find the "maximal cutter" without clearly stating why those cutters are biggest. However, whether or not a cutter is feasible for a target region can be defined based on a number of criteria; and based on those different criteria, different maximal cutters can be obtained. In order to make our approach clearer, we describe the following three alternative concepts of cutter feasibility.
   - *Alternative 1. Feasibility Definition Based on Tool's Ability to Cover the Target region*: Tool *T* is *feasible* for target region *P* if, for every point *(u,v)* in *P*, there is a permissible location *(x,y)* for *T* such that *(u,v)* is in *R(x,y)*. An example of the maximal cutter based on this definition is shown in Figure 3(a).
   - *Alternative 2. Feasibility Definition Based on Existence of Continuous Tool Path between Every Pair of Points within the Target Region*: Tool *T* is *feasible* for target region *P* if, for every pair of points *(u,v)* and *(u',v')* in *P*, there is a continuous tool path *L* such that every point *(x,y)* in *L* is a permissible location for *T* and the covered region of *L* contains *(u,v)* and *(u',v')*. An example of the maximal cutter based on this definition is shown in Figure 3(b).
   - *Alternative 3. Feasibility Definition Based on Passing Through All Bottleneck Segments*: Tool *T* is *feasible* for target region *P* if *T* can pass through all possible *bottleneck segments* formed by the Voronoi diagram of *P*.[1] An example of the maximal cutter based on this definition is shown in Figure 3(c).
4. *Maximal Tool*: Tool T_m is *maximal* for target region *P*, if and only if T_m is feasible for *P*, and any tool *T* larger than T_m is not feasible for *P* (refer to Figure 4 for a graphical illustration). The size of T_m will depend on which alternative of the feasibility definition is used (more on this later).

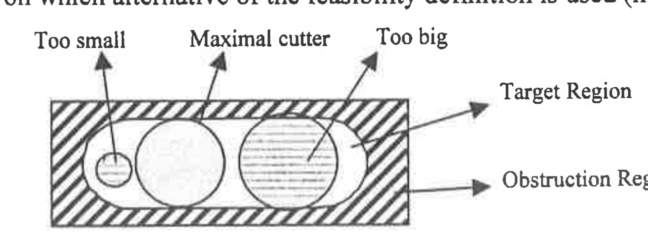


Figure 4: Maximal Tool

5. *Sweeping* operation: The *sweeping operation* is the operation in which we move a circle along a closed edge or an obstruction edge on its non-material side tangentially. Figures 5(a) and 5(b) show an example.

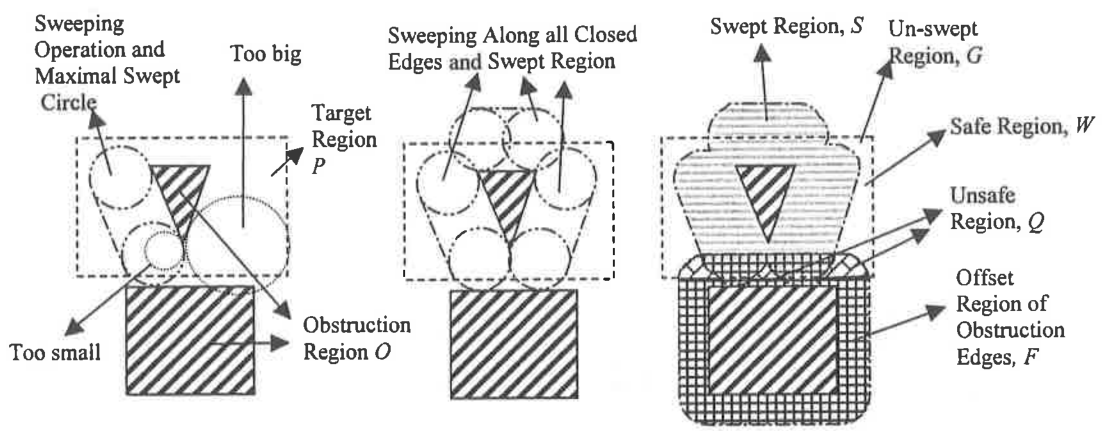


Figure 5: Maximal Sweep Circle, Swept, Unswept, Safe and Unsafe Regions

6. *Maximal Swept Circle* for Closed Edges: The *maximal swept circle* for a closed edge is the maximal tangent circle that can be moved in the non-material side of the edge without interfering with any obstruction. An example of the maximal swept circle is shown in Figure 5(a).

7. Suppose we are given a cutter T, a target region P and an obstruction region O. Then:

- *Swept region*: The swept region S is part of the target region that will be covered by sweeping a tool along all closed edges. For examples, see Figure 5(b, c).
- *Unswept region*: The *unswept region* G is the part of the target region that will not be covered by sweeping the tool along the closed edges (see Figure 5(c)). G is given by

$$G = P -^{*} S \tag{3}$$

- *Unsafe region*: The *unsafe region* Q is the part of the unswept region G such that the shortest distance from every point in that region to the obstruction boundary is smaller than the radius of the cutter used for the sweeping operation (see Figure 5(c)). If F is the offset region using r along all obstruction edges, then the unsafe region is given by

$$Q = F \cap^{*} G \tag{4}$$

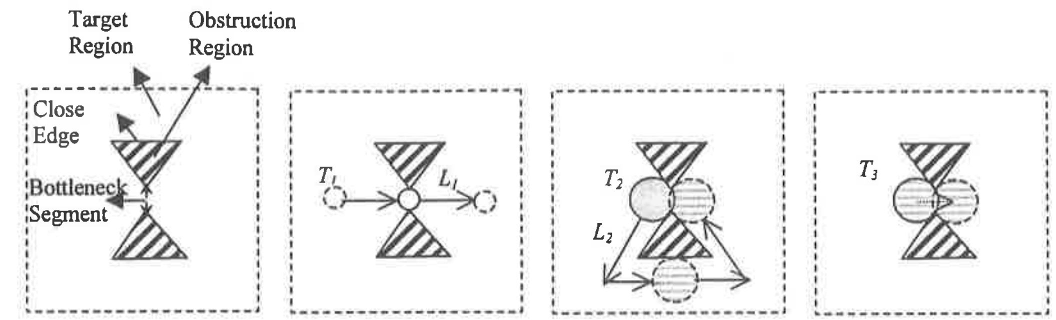
- *Safe* region: The safe region W is part of the target region such that the distance from every point in W to the obstruction boundary is greater than r (see Figure 5(c)). W is given by

$$W = G -^{*} Q \tag{5}$$

OVERVIEW OF OUR APPROACH

Basic Idea behind Our Approach

Most algorithms in the existing literature are based on alternative 3 of our cutter feasibility definition. It is easy to think of taking the minimum of all possible "bottleneck segments" generated by the Voronoi diagram of target region P (for example, see Figure 6(a)), and then using this minimum distance as the diameter of the cutter. But in many milling operations we can get the required geometric form without forcing the cutter to go through *every* bottleneck (Figure 6(b)). We



(a) Target and Obstruction Regions (b) Cutter goes through the bottleneck. (c) Cutter goes a different path. (d) Cutter jump over the bottleneck.

Figure 6: Different Cutter Path with Different Maximal Cutter

can allow the cutter either to take a different path (Figure 6(c)), or to jump over the bottleneck by lifting the cutter up and placing it across the bottleneck (Figure 6(d)).

The main goal of finding the maximal cutter is to save the manufacturing time and therefore reduce manufacturing cost. Generally, the time spent in cutting by using a small cutter is much longer than using a bigger cutter along with some lifting operations or an alternative path. Hence, we believe that feasibility definition based on the alternative 1, is an attractive candidate for cutter selection and cutter path generation. We use feasibility definition based on alternative 1 in our research work.

Algorithm for Finding Maximal Cutter

We define the cutter selection problems as following: given a target region P and an obstruction region O, find the largest cutter T that can cover P. As described below, the main algorithm is called ***Find_Maximal_Cutting_Tool.*** For every closed edge A, it calls the algorithm ***Maximal_Swept_Circle_For_Edge*** to find the maximal swept circle T. Then we use the smallest of those maximal swept circles (denoted by T_s) to sweep along all closed edges in order to split the whole region into a swept region and an unswept region. We can divide the unswept region into safe and unsafe regions using the equations in Section 2.2. For the unsafe region, we call the algorithm ***Maximal_Cover_Circle_For_Point*** to find the maximal cutter T_u that can cover it. The final result is the smaller one of T_s and T_u.

Procedure *Find_Maximal_Cutting_Tool(P, O)*
//P is the target region, and O is the obstruction region.

1. Initialize $d_l = \infty$;
2. For each closed edge A and obstruction edge C:
   - d' = ***Maximal_Swept_Circle_For_Edge****(A,C)* (see Section 4 for details);
   - $d_l = \min\{d', d_l\}$;

   //d_l is the diameter of the maximal swept circle for all closed edges.
3. Let S be the swept region produced by using the cutter with diameter of d_l performing a sweeping operation along all closed edges. More specifically, let $S = \underset{e \in E_c}{\cup^{*}} \{T_e\}$, where E_c is the set of all closed edges, and T_e is the swept region generated by sweeping T along a closed edge e;
4. Let $G = P -^{*} S$ be the unswept region;
5. Let F be the offset region of all obstruction edges by using the radius of T. More specifically, let $F = \underset{e \in E_o}{\cup^{*}} \{F_e\}$, where E_o is the set of all obstruction edges, and F_e is the offset region of an obstruction edge e;

6. Divide G into safe region W and unsafe region Q. We can get Q by taking the regular intersection of the offset region F and the unswept region G, as shown in Equations 2.4 and 2.5;
7. While Q is not empty, do:
   - Arbitrarily choose a point p, $p \in Q$;
   - d'' = ***Maximal_Cover_Circle_For_Point***(*p*, *P*, *O*) (see Section 5 for details);
     //find the maximal circle d'' that can include p and does not interfere with O.
   - $d_2 = \min\{d'', d_2\}$;
   - $Q = Q - Q_n$, where Q_n = the region covered by d_2;
   - repeat;

   //find the maximal circle d_2 that can cover the complete unsafe region.
8. Return $d = \min\{\, d_1 , d_2 \,\}$;

FINDING THE MAXIMAL CUTTER FOR A CLOSED EDGE

In the algorithm ***Find_Maximal_Cutting_Tool***, the purpose of the subroutine ***Maximal_Swept_Circle_For_Edge*** is to solve the following problem: given a closed edge A and an obstruction edge C, find the largest circle Y that can be swept along A without intersecting C.

There are several possible cases, as shown in Figures 7 and 8. In these figures, the closed edge A, with end points p_1 and p_2, is the edge along which the circle Y (of diameter d) should be swept tangentially. The obstruction edge C', with end points q_1 and q_2, is the edge that Y cannot intersect during the sweeping operation. As defined in Section 2, each closed/obstruction edge has a material side and a non-material side. We define the positive normal direction N of A to be the direction vector that is perpendicular to A and points toward the non-material side of A. We say an obstruction edge C is *on the material side* of a closed edge A if and only if every point on C is in the material side of A. As shown in Figure 7, if A is a line segment, we draw two infinite rays, l_1 and l_2, that are perpendicular to A along N, pointing to the non-material side of A and starting from each end point of A. As shown in Figure 8, if A is an arc segment, we draw two infinite rays, l_1 and l_2, from each end point radially, pointing to the non-material side of A. In both cases, l_1 and l_2 will divide the whole non-material side of A into three sub-regions. We call those regions *I, II, III*, as shown in Figures 7 and 8.

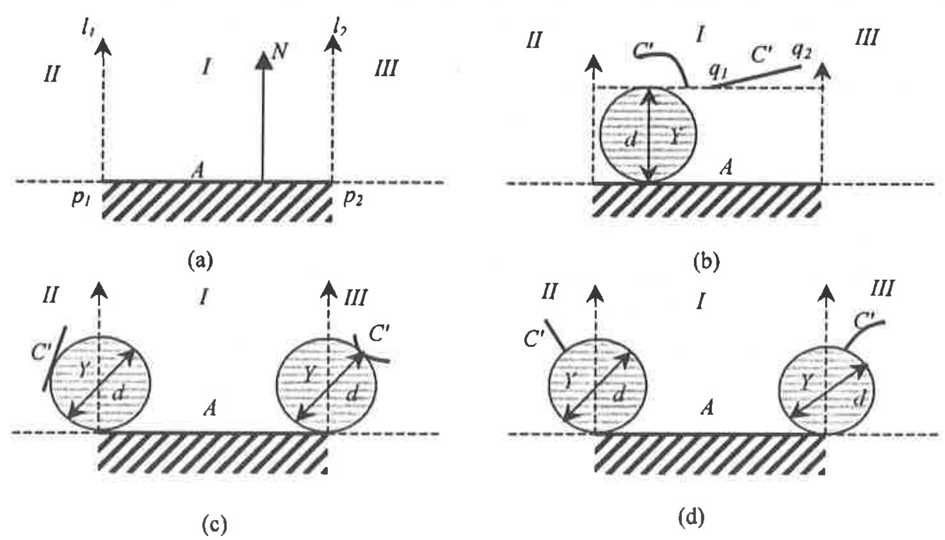


Figure 7: Finding the maximal cutter when the closed edge A is a line segment.

Procedure ***Maximal_Swept_Circle_For_Edge*** *(A, C)*
1. Split C into segments such that each segment is completely contained in regions *I, II, III*, or the material side of A;
2. $d = \infty$;
3. For every segment C' of C, do the following:
   - If C' is on the material side of A then we don't need to consider C', therefore $d' = \infty$;
   - Else:
     - If A is a line segment, d' = ***Maximal_Swept_Circle_For_Line***(*A,C'*);
     - If A is an arc segment, d' = ***Maximal_Swept_Circle_For_Arc***(*A,C'*);
   - $d = \min\{d,d'\}$;

Procedure ***Maximal_Swept_Circle_For_Line***(*A,C'*)
1. If C' is in region *I*, return d = distance between A and C';
   //Because in this region, when the circle sweeps along A, the distance between A and C' will be
   //the maximal diameter of the circle. An example is shown in Figure 7(b).
2. If C' is in region *II* or *III*:
   - Try to find a circle Y that is tangent to both C' and A at one of the end points of A and the center of Y is located on l_1(for region *II*) or l_2(for region *III*), then return d={the diameter of Y};
     //An example is shown in Figure 7(c).
   - If such a circle does not exist, then construct two circles passing through two end points of C' such that centers of circles are located on l_1 or l_2, and circles are tangent to A at the one of its end points. Return the diameter of the smaller circle.
     //An example is shown in Figure 7(d).

Procedure ***Maximal_Swept_Circle_For_Arc***(*A,C'*)
1. If C' is in region *I*, return d= the distance of A and C'};
   //Because in the region, when the circle sweeps along A, the distance between A and C' will be
   //the maximal diameter of the circle. An example is shown in Figure 8(b).
2. If C' is in the region *II* or *III*, from each end point of A draw two rays tangent to A and pointing to the non-material side of A, called B, B':
   - Try to find a circle Y that is tangent to both C' and B (or B') at the start point of B (or B') and the center of Y is located on l_1 (for region *II*) or l_2 (for region *III*), then return d={the diameter of Y};
     //An example is shown in Figure 8(c).
   - If such a circle does not exist, then construct two circles passing through two end points of C' such that circle centers are on l_1 or l_2, and they are tangent to B (or B') at the start point of B (or B'). Return the diameter of the smaller circle.
     //An example is shown in Figure 8(d).

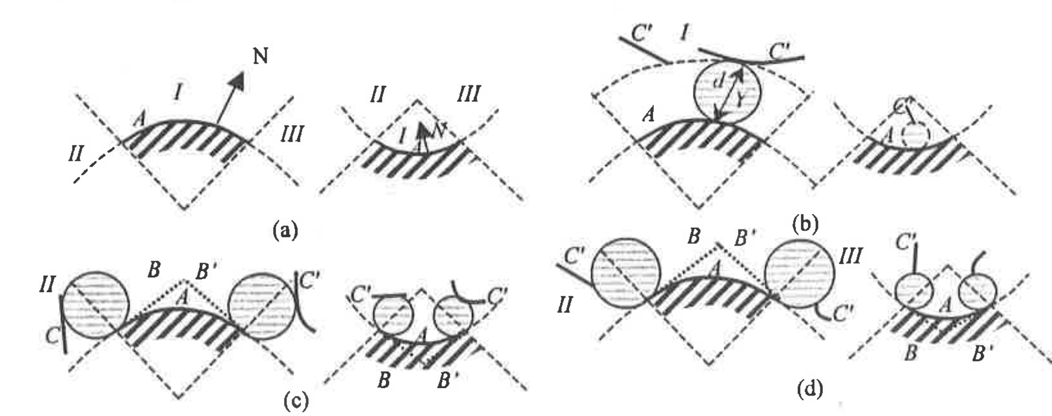


Figure 8: Finding the maximal cutter when the closed edge is an arc segment.

FINDING THE MAXIMAL CUTTER FOR UNSAFE REGION

In the algorithm ***Find_Maximal_Cutting_Tool***, the purpose of the subroutine ***Maximal_Cover_Circle_For_Point*** is to solve the following problem: given a point (x_a, y_a) and a set of obstruction edges, find the maximal circle T such that (x_a, y_a) is contained in T, and T does not intersect with any obstruction edges. Figure 9 gives an example.

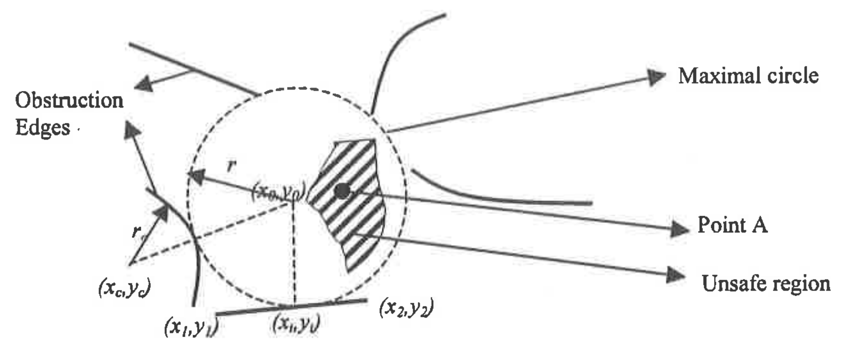


Figure 9: Maximal Cover Circle

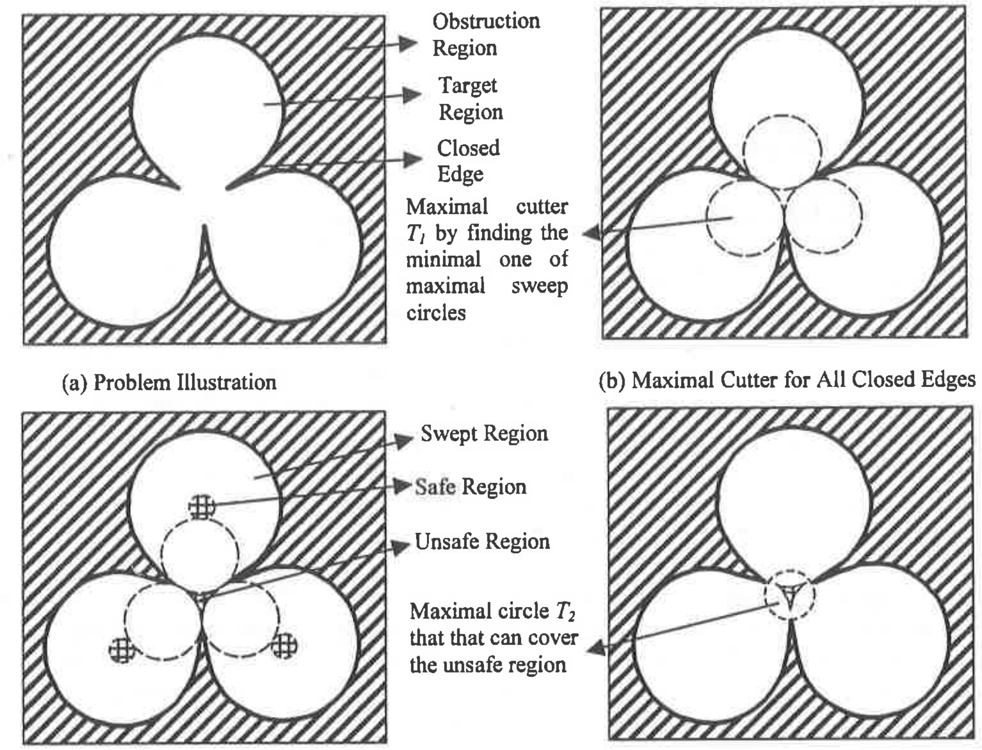
We formulate this problem as a non-linear optimization problem.[7] T is the circle with center (x_0, y_0) and radius r. The variables for the optimization problem are x_0, y_0 and r. The objective is to maximize r, subject to the following constraints:

1. The circle T should include the point A, i.e., $(x_a - x_0)^2 + (y_a - y_0)^2 \le r^2$;
2. The circle T should not intersect any linear obstruction edge: Suppose l is a linear obstruction edge with end points (x_1, y_1) and (x_2, y_2); let l' be the line containing e; let v be the line perpendicular to l' that contains (x_0, y_0); let (x_i, y_i) be the intersection point of l' and v; and let d be distance from (x_0, y_0) to l'. If (x_i, y_i) is in l (i.e., $d \ge r$), then we need the following constraints to eliminate the intersection between the circle and the obstruction edge: $(x_1 - x_0)^2 + (y_1 - y_0)^2 \ge r^2$; $(x_2 - x_0)^2 + (y_2 - y_0)^2 \ge r^2$.
3. The circle T should not intersect any arc obstruction segment: Suppose a is an arc obstruction edge with end points (x_1, y_1), (x_2, y_2), radius r_c, and center point (x_c, y_c); let a' be the circle containing a; let l be the line containing (x_c, y_c) and (x_0, y_0); let (x_i, y_i) be the intersection point of a' and l; and let d be distance from (x_0, y_0) to a'. If (x_i, y_i) is in a (i.e., $d \ge r$), then we need the following constraints to the eliminate intersection between the circle and the obstruction edge: $(x_1 - x_0)^2 + (y_1 - y_0)^2 \ge r^2$; $(x_2 - x_0)^2 + (y_2 - y_0)^2 \ge r^2$.
4. The circle's radius should be non-negative: $r \ge 0$.

IMPLEMENTATION AND EXAMPLES

We use the example shown in the Figure 10 to illustrate the operation of our algorithm. The target region and obstruction regions are shown in Figure 10(a). The details are as follows:

- First we need to find the maximal circle T_1 that can be swept along all closed edges (shown in Figure 10(b)). Then we use T_1 to perform sweeping along all closed edges to divide the whole region into two regions: the swept region, and the unswept region, as shown in Figure 10(c).



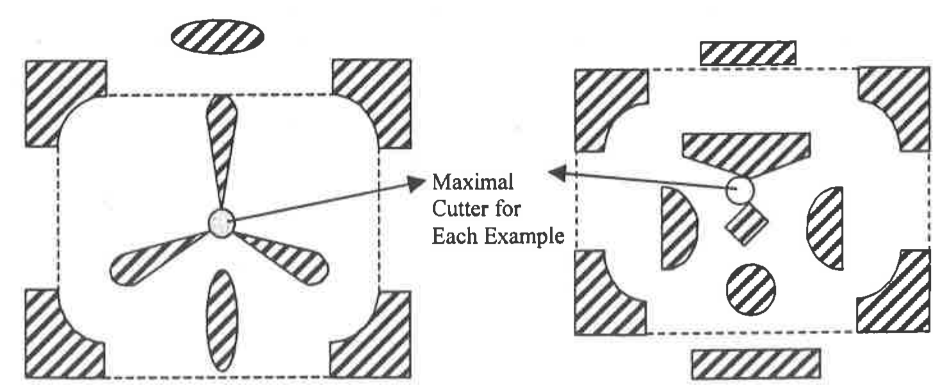
(a) Problem Illustration (b) Maximal Cutter for All Closed Edges

(c) Sweeping Operation (d) Maximal Cutter for Unsafe Region

Figure 10: An example of the operation of our algorithm.

- In the unswept region, there are two types of sub-regions, i.e., the safe region and the unsafe region. For the unsafe region, we should find the maximal circle T_2 that cover it and does not interfere with any other edges on the obstruction boundary, as shown in Figure 10(d).
- The maximal circle that covers the target boundary without interfering with the obstruction boundary will be the minimal one of T_1 and T_2.

Some other examples are shown in Figure 11. In Figure 11(a), after performing the sweeping operation, we still need to find the maximal circle that covers the unsafe region before we can obtain the final result. But in Figure 11(b), there is no unsafe region left after the sweeping operation, we can obtain the final result directly.



(a) (b)

Figure 11: Other Examples

CONCLUSION AND DISCUSSION

In this paper, we have presented a geometric algorithm for finding the maximal cutter size for a 2-*D* milling process. Our algorithm has the following properties:

- It finds the largest cutter that can cover the region to be machined without interfering with the obstruction regions.
- In addition to solving traditional pocket-milling problems, our algorithm can solve a wide variety of milling problems that involve open edges. Consideration of open edges is extremely important when near-net shape castings are used as starting stocks.
- Our algorithm uses a cutter feasibility definition based on cutter's ability to cover the target region. Therefore, it can find larger cutters than the ones found by algorithms that are based on alternative definitions of feasibility (e.g., either based on covering every bottleneck in the target region or existence of continuous path between every pair of points in the target region).
- We believe that the algorithm presented in this paper is admissible for most real life engineering problems; i.e., if a feasible cutter exists, then the algorithm will return the diameter of the largest possible feasible cutter. We are currently working on the classification of 2-D milling problems for which this algorithm is admissible.

We are currently developing an implementation of our algorithm. Our current implementation only works with target and obstruction regions with linear edges (for example, the target region shown in Figure 6), but we are extending our implementation to handle circular edges. In our future work, we plan to (1) use Voronoi diagrams for computing the maximal cover circle as an alternative to using non linear programming, and (2) perform cutter selection optimization for multiple cutters.

ACKNOWLEDGEMENT

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. We would like to thank Wei Qiu for implementing a preliminary version of the cutter selection algorithm.

REFERENCES

1. 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.
2. 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.
3. M. Bala and T. C. Chang, Automatic cutter selection and optimal cutter-path generation for prismatic parts, *International Journal of Production Research*, 1991, 29(11), 2163-2176.
4. 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.
5. Y. S. Lee, T. C. Chang, Using virtual boundaries for the planning and machining of protrusion free-form features, *Computers in Industry*, 25(1994) 173-187.
6. 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.
7. Singiresu S. Rao, Engineering Optimization-Theory and Practice, third Edition, A Wiley-Interscience Publication, 1996

MECHANICAL MACHINING OF AEROSPACE TITANIUM ALLOYS WITH ULTRA-HARD CUTTING TOOLS

FARHAD NABHANI
University of Teesside, School of Science & Technology,
Middlesbrough, TS1 3BA, UK

ABSTRACT: The performance of PCBN (AMBORITE*) and PCD (SYNDITE) has been compared with that of coated tungsten carbide tool currently being used to machine titanium aerospace alloy. Tests confirm that SYNDITE gives a better surface finish, longer tool life and more manageable swarf than other tools. In addition, the 'quick stop' technique establishes that, for all three cutting tools, a layer is formed between the rake face and the underside of the emerging chip which has a fundamental effect on cutting and wear mechanisms.

INTRODUCTION

Titanium is an attractive material to aerospace designers due to its unique combination of strength and lightness. However, it poses considerable problems in manufacturing because of its poor machinability. Traditionally high speed steel and monolithic carbide cutting tools have been employed and a relatively short lifetime or the need for frequent cutter regrinding has been accepted. More recently there has been a move towards the adoption of insert tooling based upon coated carbide systems. With the evolution of a number of new cutting tool materials there is evidence to suggest that the use of some of the ultra hard materials, such as those based on polycrystalline diamond or cubic boron nitride, may be advantageous in the machining of titanium alloys.

An experimental programme has been conducted to explore the potential of such materials by means of machining trials and by the use of a 'quick-stop' device. The cutting tool materials examined were coated carbide specification typical of those used by aerospace manufacturers to machine titanium alloys, a high cubic boron nitride content tool material (AMBORITE) and polycrystalline diamond (SYNDITE).

EXPERIMENTAL DETAILS AND OBSERVATIONS

Material specifications

The workpiece material used throughout all these tests was an as-rolled and annealed TA48 titanium alloy with a nominal composition (in wt%) given as Al: 5; Mo: 4; Sn: 2 – 2, 5; Si: 6 – 7: Fe: 2.0 max; H: 0.015; O: 0.25; N: 0.05; Ti: remainder. It had a Knoop hardness (1.0 kg load) of 425 kg/mm^2 (4.17 Gpa) and a microstructure which consisted of elongated alpha phase in a fine dark-etching beta matrix.