Approximation Algorithm for Multiple-Tool Milling*

Sunil Arya† Siu-Wing Cheng‡ David M. Mount§

June 26, 1999

Abstract

Milling is the mechanical process of removing material from a piece of stock through the use of a rapidly spinning circular milling tool in order to form some desired geometric shape. An important problem in computer-aided design and manufacturing is the automated generation of efficient milling plans for computerized numerically controlled (CNC) milling machines. Among the most common milling problems is simple 2-dimensional pocket milling: cut a given 2-dimensional region down to some constant depth using a given set of milling tools. Most of the research in this area has focused on generating such milling plans assuming that the machine has a tool of a single size. Since modern CNC milling machines typically have access to a number of milling tools of various sizes and the ability to change tools automatically, this raises the important optimization problem of generating efficient milling plans that take advantage of this capability to reduce the total milling time.

We consider the following *multiple-tool milling problem*: Given a region in the plane and a set of tools of different sizes, determine how to mill the desired region with minimum cost. The problem is known to be NP-hard even when restricted to the case of a single tool. In this paper, we present a polynomial-time approximation algorithm for the multiple-tool milling problem. The running time and approximation ratio of our algorithm depend on the simple cover complexity (introduced by Mitchell, Mount, and Suri) of the milling region.

1 Introduction

Milling is one of the most important methods used in the manufacturing of mechanical parts in computer-aided manufacturing (CAM). It is applied to a

*A preliminary version of this paper appeared in the *Proc. of the 14th Annual ACM Symposium on Computational Geometry*, 1998, pp. 297–306.

†Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Research supported in part by RGC CERG HKUST 736/96E and RGC CERG HKUST 6158/98E.

‡Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Research supported in part by RGC CERG HKUST160/95E.

§Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland. Research supported in part by the National Science Foundation under grant CCR–9712379.

workpiece of (typically metal) stock sometimes called the *part* or *billet*, which is clamped to a moving platform that is then translated under a rapidly spinning circular-shaped *milling tool*. It is somewhat more natural to think of the stock as remaining stationary and the tool translating above it. In this way material is removed, or *milled*, from the part. The overall problem is how to construct milling plans in order to achieve a given final geometric shape within the shortest amount of time.

There are several kinds of milling depending on the numbers of degrees of freedom possessed by the tool relative to the workpiece. In this paper we focus on the simplest case, but one that is common in practice, where continuous tool movement is possible in one plane and the direction normal to it is used only for retracting the tool. This situation is commonly referred to as 2D milling or *pocket machining*. (Pocket refers to the region being milled).

There has been a lot of research on the subject of automatic generation of tool paths for computerized numerically controlled (CNC) pocket machining. However, much of this study, both theoretical and practical, has focused on machining pockets using a single tool; the question of how to machine pockets efficiently using more than one tool has been largely ignored, and seems to be considerably deeper and richer than the single-tool problem. Modern milling machines have the capability of automatically loading different milling tools of a wide range of radii. Using a larger tool when possible offers a significant advantage in terms of milling time. In this paper we propose a cost model for describing multiple-tool milling problem, and present an approximation algorithm.

Previous results. Within the computer-aided design and manufacturing community there has been a considerable amount of study of various heuristics for the automatic generation of tool paths for pocket machining. The most common general strategies are *contour-parallel* (also known as *window-pane milling*) [Bru82, HA92, Per78, Pre89, PK83, PK85, SL90], in which the tool spirals inwards from (or outwards to) the boundary of the region, and *axis-parallel milling* (also known as *zig-zag* or *staircase milling*) [Bru82, DBW93, Kra92, PGW90, W$^+$87], in which the milling tool moves back and forth cutting parallel strips. Multiple tool milling has been considered, see for example [BC91], but there is no theoretical analysis of the performance of the heuristics proposed.

Held [Hel89, Hel91a, Hel91b] made a comprehensive study of milling heuristics from a computational geometry perspective. For the single tool case, he presented efficient algorithms to find a feasible tool path, given the shape of the pocket to be milled and the size of the tool. On the theoretical side, Arkin, et al. [AFM93, AFM95] and Iwano, et al. [IRT94] have given constant-factor approximation algorithms for finding shortest paths for the single-tool milling problem and for the closely related problem of lawnmowing. Arkin, et al. [AHS96] have also given approximation algorithms for minimizing the number of retractions for the zig-zag milling problem, subject to the constraint that one is not allowed to mill the same region again. The problem is known to be NP-hard even when restricted to the case of a single tool [AFM93]. We know of no theoretical work considering the use of multiple tools in milling.

Domain and tools. We model the pocket machining problem as follows. Tools are changed at a designated location called the *tool-change center*. Thus the path for each tool is assumed to start and end at this center. The input to our problem provides a planar domain P to be milled, a set of tools of different sizes, and the location of the tool-change center. We make the realistic assumptions that the ratio between consecutive tool sizes is bounded above by a constant (for simplicity, we assume this ratio to be bounded by 2), and that the smallest tool can mill P without the need to be lifted. These two assumptions are essential in proving the approximation ratio. (We leave the removal of these assumptions as future research problems.) The tools are disks with different radii and the domain P is a connected region (possibly with holes) bounded by straight line segments and circular arcs. We call these segments and arcs *domain edges*. We assume that each domain edge has two distinct endpoints. The tools can be moved arbitrarily, as long as they do not cross the boundary of P.

Let n denote the number of vertices in P and let m denote the number of tools.

Cost model. A milling plan is a sequence of tours, each for a particular tool size. A tour begins and ends at the tool-change center. It consists of a sequence of paths alternating between being engaged with the material (milling path) or being retracted (transport path in air). Due to stress on the tool, the speed with which the tool can be moved, called the *feed rate* is typically much smaller for milling than for transport. Consider a milling plan Ψ. Define

$$\begin{aligned} mill(\Psi) &= \text{total milling path length for } \Psi, \\ transport(\Psi) &= \text{total transport path length for } \Psi, \\ ntools(\Psi) &= \text{the number of tool changes in } \Psi. \end{aligned}$$

The total milling cost in this model is

$$cost(\Psi) = \alpha \cdot mill(\Psi) + \beta \cdot transport(\Psi) + \gamma \cdot ntools(\Psi),$$

where α, β, and γ are arbitrary nonnegative values supplied by the user as part of the input. Each milling path with tool of size t includes a cost of $2t$ in addition to the length of the path. This additional component is included to account for the time to place the cutting tool within the material. This might be done either by milling in from the side or by drilling a hole and milling down into the material. This assumption is added to prevent ridiculous solutions based on using the tool like a cookie-cutter to stamp out disks without paying any milling cost at all. From a practical standpoint, plunging the tool into material induces considerable stresses on the milling tool, and is not used in practice, or only after the time has been spent to drill a hole where the center of the milling tool is to be placed.

The factor γ reflects the amount of time needed to load a tool. We include this cost when loading the first tool. Note that under our cost model, there is no advantage gained by loading a tool, unloading it, and then reloading it later. Thus it is reasonable to assume that each tool is loaded at most once, and hence *ntools* is equal to the number of tools used by Ψ.

Output representation. As observed in [AFM95], the milling path of a tool may require a combinatorially very large description even if the size of the

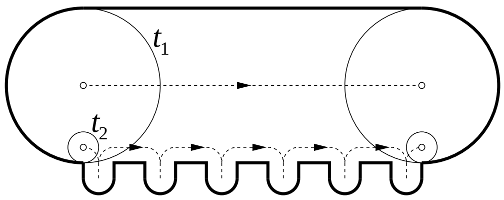

Figure 1: Counterexample for the simplest milling strategy.

region milled is combinatorially very small, e.g., a small tool milling a large circle. Following the approach in [AFM95], we use a succinct representation of milling paths instead. In our output, we represent the points milled by each tool as a collection of simple regions of regular structure. If desired, the actual milling paths can be extracted by contour-parallel milling or zig-zagging within each output regions. We also output the cost of our approximate milling plan.

2 Overview and summary of results

Before discussing our approximation algorithm, we begin with some discussion to motivate the various elements of our solution. Since large tools can mill more material per unit of motion than small tools, the simplest strategy that comes to mind is to mill everything that can be reached by the largest tool, and then repeatedly load successively smaller tools and mill everything that is reachable for each tool. However, it is easy to see that this simple strategy may be suboptimal by a factor that is as large as the number of different tools. For example, for the domain shown in Fig. 1, after the large tool t_1 has acted, all that remains are the small protrusions. The next smaller tool may only be able to shave away a small amount of additional material. The best option is to load a much smaller tool t_2 that can completely fit within each of the small protrusions. The tradeoff that must be faced is whether to use a larger tool and mill potentially less material with greater efficiency, or to use a smaller tool and mill more material with lesser efficiency.

At a very abstract level, milling the domain is equivalent to covering the points in the domain with copies of the tools available. Each copy of a tool used will incur some cost. This cost includes the time to load the tool, the time to mill the various regions, and the time to transport the tool from one unmilled region to the next. This suggests that milling is related to the discrete optimization problem of weighted set-cover (cover a domain by sets, each having an associated cost, so that the sum of costs is minimized). A well-known heuristic for weighted set-cover is the greedy algorithm [Chv79], which at each stage selects the subset that maximizes the number of items covered per unit cost. This algorithm is known to produce a logarithmic approximation ratio.

We will transform the multiple-tool milling problem into a weighted set-cover problem and then solve the weighted set-cover problem by a greedy heuristic. To construct the transformation, we need to define the elements in the base set and the weighted subsets. The transformation is not straightforward for two reasons. First, it is infeasible to use points as set elements directly as there are an infinite number of them. We overcome this by discretizing the domain into simple regions and use these simple regions as set elements instead. We will show how to construct this discretization such that we may assume that each simple region is milled with only one tool, while increasing the approximation ratio only by a constant. This is given in Sections 4 and 5.

Each subset in our transformation will correspond to a milling action, which consists of loading a tool and then milling some subset of the remaining unmilled regions with this tool. The second problem is that it is not efficient to enumerate the exponential number of possible subsets of unmilled regions in order to select the next subset. We overcome this by using an approximate greedy strategy that does not require the weighted subsets to be explicitly provided. This strategy will incur another constant factor in the approximation ratio, and it is based on the Euclidean k-TSP problem [Aro97]. It will be described in Section 6.

Our discretization of milling actions is based on a subdivision of the milling domain P. We first subdivide the boundary of P through the introduction of new vertices into $O(n)$ segments, in order to satisfy certain monotonicity conditions, which will be described later. Let P^* denote the modified domain. The size of our discretization is equal to the simple cover complexity of P^*. The *simple cover complexity*, or $scc(P^*)$, is an intrinsic measure of the geometric complexity of P^* [MMS94]. It is defined as follows. A disk is *simple* if it intersects at most 2 edges of P^*. Given any $\epsilon > 0$, we say that a ball of radius r is *ϵ-strongly simple* if the ball with the same center and radius $(1+\epsilon)r$ is simple. Given ϵ, a *strongly simple cover* of a region P^* is a collection of ϵ-strongly simple balls whose union contains P^*. Given any fixed ϵ (for example $\epsilon = 1/2$), the simple cover complexity of P^* is defined to be the cardinality of the smallest strongly simple cover of P^*. Our main result is:

Theorem 2.1 *Given a domain P of n vertices and m circular tools, an $O(\log m + \log scc(P^*))$-factor approximation to the optimum cost milling plan for P can be computed in time that is polynomial in n, $scc(P^*)$ and m. (Constant factors hidden by the "big-Oh" do not depend on the cost model parameters.)*

Throughout, we will denote $scc(P^*)$ by N for simplicity.

The remainder of the paper is organized as follows. In Section 3 and 4, we show how to discretize the problem. Section 5 shows that our discretization method indeed approximates the milling problem to within a constant factor. Section 6 shows how to reduce the milling problem to a weighted set cover problem and describes the approximation algorithm.

3 Subdividing the domain

Let ∂P denote the boundary of the domain. Consider the Voronoi diagram of ∂P. We define a distance function $v(p)$ which maps every point p on the Voronoi diagram to its closest point on ∂P. Consider the set of points p on the Voronoi diagram such that $v(p)$ is locally minimal, in the sense that in every sufficiently small neighborhood of p there is a point on the Voronoi diagram with strictly higher distance value, and no point in the neighborhood has a strictly smaller distance value. For each such point p on the Voronoi diagram, the two nearest points on ∂P to p are called *bottleneck points*, and the line segment joining these boundary points is a *bottleneck segment*. (See Fig. 2 for example.) Note that there cannot be three or more bottleneck points for p by the local minimality

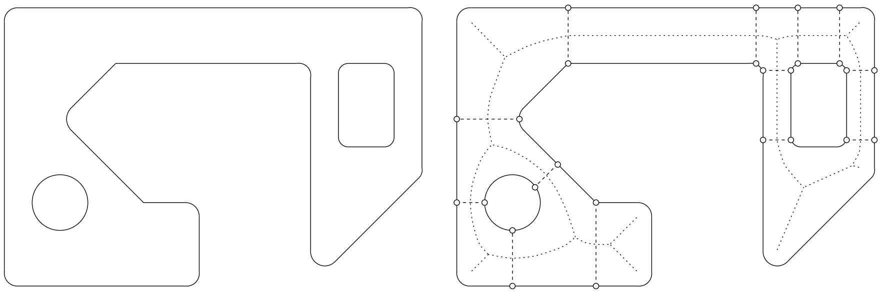

Figure 2: The dotted curve shows the Voronoi diagram of the domain. White points are the bottleneck points and dashed segments are the bottleneck segments.

requirement. We introduce all the bottleneck points as new vertices on the boundary of P. Each bottleneck point splits a domain edge into two smaller domain edges. We denote by P^* the resulting domain.

We will perform a quadtree decomposition of P^*. This decomposition will subdivide the plane into a collection of square regions called *boxes*. For any box x, we denote its side length by $width(x)$. For any positive real c, we denote by cx a box with the same center as x whose side length is $c \cdot width(x)$.

The goal of the decomposition is to cover P^* with a set X of boxes such that the portion of the domain lying within and near each box is extremely simple. X is generated as follows. We first enclose P^* in a bounding box. Then we apply the following splitting rule. For any box x, we call $3x$ the *buffer zone* of x and denote it by $buf(x)$. Take any box x, if $buf(x)$ intersects more than two domain edges, then split x through its center into four identical boxes each of half the size. When the splitting process stops (which will occur eventually since each vertex is adjacent to at most two domain edges), we obtain a set of boxes covering P^*.

A *cell* is a connected component of $P^* \cap x$ for some box $x \in X$. Given a cell C, we denote by $box(C)$ the box $x \in X$ that contains C. We define the *size* of C, $size(C)$, to be $width(box(C))$. We define $buf(C)$ to be $buf(box(C))$. The size of our subdivision of P^* is bounded by the *simple cover complexity* [MMS94]. Such a quantity has been reported to be close to linear for practical scenes [BKSV97], though hypothetical examples exist for which the simple cover complexity becomes unbounded.

Lemma 3.1 *There are $O(scc(P^*))$ boxes covering P^*.*

Proof To prove that $|X|$ is $O(scc(P^*))$ we consider an expansion factor of $\epsilon = 1/2$. As shown in [MMS94], the choice of ϵ only affects the constant factor involved. Consider a strongly simple disk of radius r. This means that its expansion by a factor of $1 + \epsilon = 3/2$ does not intersect more than two edges of P^*. We claim that any box in X that overlaps the (unexpanded) disk has width $\geq r/(12\sqrt{2})$. Suppose to the contrary that there is a box x that overlaps the disk and has width $< r/(12\sqrt{2})$. Then the parent box p of x has width $< r/(6\sqrt{2})$. Thus, $buf(p)$ has width at most $r/(2\sqrt{2})$ and hence a diameter of at most $r/2$. Thus $buf(p)$ lies entirely within the expanded disk and so intersects at most two edges of P^*. Consequently p could not have been split further to

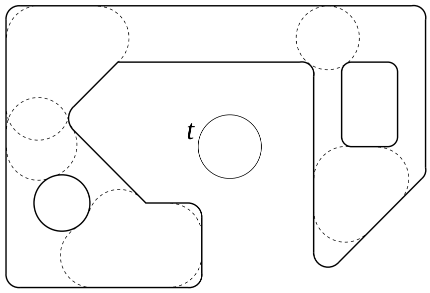

Figure 3: The accessibility arcs for tool t (dashed).

generate x. This is a contradiction. It follows that any strongly simple disk cannot contain a box of X of width less than $r/(12\sqrt{2})$. By a simple packing argument, it follows that the number of boxes in X that overlap any strongly simple disk is a constant. Since the simple disks cover P^*, it follows that $|X|$ is bounded above by a constant factor times the number of simple disks, and hence is $O(scc(P^*))$. $\square$

4 Basic milling actions

As mentioned above, our approximation algorithm is based on discretizing the space of possible milling actions into what we call *basic milling actions*, or *BMA*s for short. Each basic milling action is responsible for milling a certain portion of the domain by a single tool. In general, many tools may be able to access a given region, and the definition of BMA makes no attempt to limit which tool is responsible for some region. It will be the responsibility of the greedy algorithm, described in Section 6, to determine which BMAs to apply to employ for the final plan.

We begin by introducing some notation. Let $d(p, r)$ denote a disk of radius r centered at a point p. Given a tool t, we also use t to denote its radius. Whenever we put the center of t at a point p, we call $d(p, t)$ a *placement of* t. A placement of t is *free* if it lies within P^*. The *free space* of t is the locus of centers of all free placements of t. We denote it by $\mathcal{F}(t)$. Formally, $\mathcal{F}(t) = P^* \ominus t$, where $\ominus$ is the Minkowski difference operator. Thus, $\mathcal{F}(t) \oplus t$ is the set of points in P^* that can be covered by a free placement of t, where $\oplus$ is the Minkowski sum operator.

At each vertex p of $\mathcal{F}(t)$, $d(p, t)$ touches ∂P^* at several points. If the arc between two consecutive contact points on the boundary of $d(p, t)$ is less than a semicircle, then we call it an *accessibility arc*. The collection of accessibility arcs for a tool t is denoted by $\mathcal{A}(t)$. See Fig. 3.

A subset K of $\mathcal{F}(t)$ and a subset $M \subseteq K \oplus t$ define a *milling action*. Specifically, t moves with center in K to remove all points in M. If K is not connected, then we have to pick t up transport it to another component and place it. This will incur a charge of $2t$ per connected component in K plus the transport cost to visit all components in K. We say that a point in M is *milled* by this action. Note that, physically speaking, a larger set than M may be removed by t's movement. However, we only consider points in M as milled, and we will have to deploy other milling actions to mill points not in M. Imagine that an arbitrary sequence of milling actions (possibly of different tools) has been applied to the collection of cells defined in the previous section. Given a cell C, we call the set of points in C not milled the *unmilled region*, and we call a connected component of the unmilled region an *unmilled component*.

Given a tool t, we restrict ourselves to two kinds of milling actions of t on a cell C, depending on the relative sizes of t and C. We say that t is *large* for a cell C if $t \geq size(C)/16$ and *small* if $t \leq size(C)/4$. Conversely, we say that a cell C is *small* for t if $t \geq size(C)/16$ and *large* if $t \leq size(C)/4$. Note that this implies that in the range $size(C)/16 \leq t \leq size(C)/4$, t is both small and large for cell C. In the next two subsections we describe the basic milling actions for large tools and small tools.

4.1 Large-tool basic milling action

Recall that the goal of defining basic milling actions is to provide a discretization within which to approximate any milling action. Since we do not know the optimum milling path for any tool, our approach will be to define each large-tool basic milling action to mill a local region whose size is proportional to the size of the milling tool. Then we will be able to approximate the milling action of any milling plan by concatenating a sequence of such local milling operations. Since basic milling actions are defined independently of one another, we cannot generally predict whether a given milling action will simply be a continuation of a neighboring milling operation, or whether it will require placing the tool of size t into the material. Recall that each such placement incurs a cost of $2t$ by our model. To absorb this potential placement cost, we define each basic milling operation (except for the smallest tool) so that there is a free placement of a tool of twice this size. This insures that there will be sufficient millable area, so that placement costs will not dominate milling costs.

One of the tricky issues in defining basic milling actions for large tools is that a single large tool may mill portions of many small cells at the same time. Thus, if we were to account for the milling cost on a cell-by-cell basis and sum these costs, then we may considerably overestimate the actual milling cost. In order to accurately account for the total cost of using a large tool to mill many smaller cells, it is important to define the milling actions for large tools in a way that is global to the small cells that it affects. We do this by overlaying a grid on the domain whose side length is proportional to the tool size, and then associating each basic milling action with each grid cell. A second issue is predicting the possible shapes of the unmilled regions that result after each basic milling action. To minimize the number of possibilities, our milling actions are defined so that if a cell cannot be milled entirely, then we mill up to an accessibility arc for this tool.

Overlay a square grid $\mathcal{G}_t$ of side length t on P^*. Define $\mathcal{B}_t$ to be the set of grid squares b such that $b \oplus 4t$ overlaps some quadtree box of width less than $16t$. Any grid square $b \in \mathcal{B}_t$ induces a (possibly empty) set of milling actions as follows. If t is the smallest tool, let K be a connected component of $(b \oplus 28t) \cap \mathcal{F}(t)$ such that K overlaps with $b \oplus 4t$; otherwise, let K be a connected component of $(b \oplus 28t) \cap \mathcal{F}(t)$ that contains some component of $b \cap \mathcal{F}(2t)$. Define $\mathcal{C}_b$ to be the collection of cells C such that t is large for C and C overlaps $b \oplus 4t$. Define M

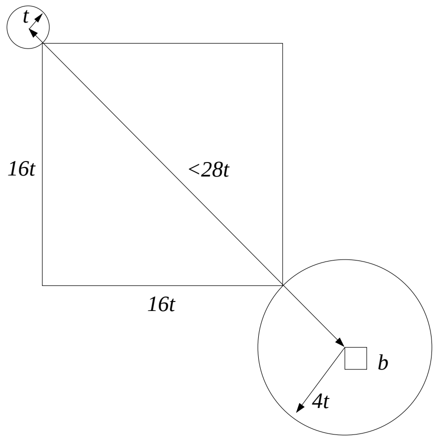

Figure 4: An illustration of the choice of the constant 28. The distance between the center of t on the upper left of the figure from the upper left corner of b is $4t + 16\sqrt{2}t + t \approx 27.7t < 28t$.

to be $K \oplus t \cap \bigcup_{C \in \mathcal{C}_b} C$. Then K and M define a *large-tool basic milling action* denoted by *large_tool*(t, K, M), i.e., move t along the surface with center in K to mill points in M.

The choice of the constant 28 is due to the following motivation. We want a large BMA to simulate the local milling action of a large tool on small cells in the optimal milling plan. Suppose that the optimal milling plan uses a tool with center in the grid square b. We want to do the simulation with a tool of radius a constant factor smaller (as will be shown, we need this to guarantee that b will induce only $O(1)$ large tool BMAs for any tool). By our assumption that the radii of two successive tools are within a factor 2, we know that we can always pick a tool which is at least a factor 2 smaller but no more than a factor 4 smaller. Hence, we use a tool t to simulate a tool of radius at most $4t$ in the optimal milling plan. As mentioned before, we need to push up to accessibility arcs to simplify the shape of unmilled regions. Thus, if a tool of radius $4t$ centered inside b mills a cell small for t , then we want the large BMA of t to cover as many points in the cell as $\mathcal{F}(t)$ allows. Recall that the cell is small for t and hence has size at most $16t$. Fig. 4 shows that $b \oplus 28t$ is large enough to allow the tool t to attack the cell from any direction. An example of the effect of a large BMA is shown in Fig. 5.

In the remainder of this section we establish a number of facts about large-tool BMAs. In Lemma 4.1 we show that after the action of a large-tool BMA by t, the boundaries of any resulting unmilled component consist of accessibility arcs of t. Lemmas 4.2 and 4.3 show that the length of the boundary of K is $O(t)$. Lemma 4.4 shows that the cost of each large-tool BMA is $O(t)$. Finally, Lemmas 4.5 and 4.6 and the associated corollary establish that there are a total of $O(mN)$ large-tool BMAs.

Lemma 4.1 *Let C be a cell small for tool t. For any large_tool(t, K, M), either $M \cap C$ is empty or the boundary of $M \cap C$ consists of points in ∂C and $\mathcal{A}(t)$.*

Proof Suppose that $M \cap C$ is nonempty. Take a point q in $\partial(M \cap C) \setminus \partial C$. By our choice, q is milled by *large_tool*(t, K, M) and so there is a free placement

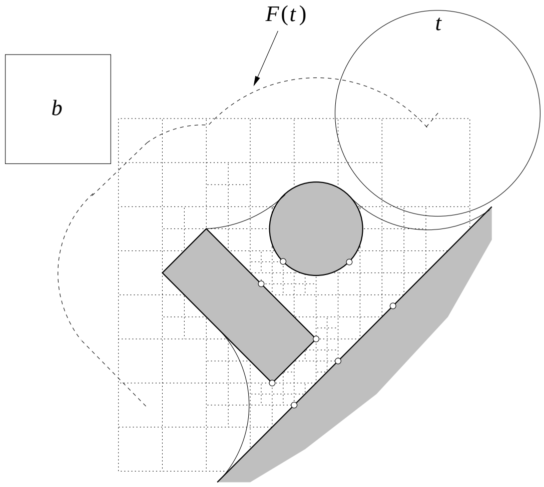

Figure 5: Large-tool BMA for tool t. The solid square is a grid square b in $\mathcal{G}_t$. The quadtree cells shown in dotted lines are within $b \oplus 4t$. All of them are small for t and so b induces large tool BMAs of t on them. After these large tool BMAs, some cells will be milled completely and some are milled up to accessibility arcs (shown in solid lines). The boundary of $\mathcal{F}(t)$ is shown in dashed lines.

$d(p,t)$ that touches q, where $p \in K$. We claim that p must lie on the boundary of $\mathcal{F}(t)$ and so $q \in \mathcal{A}(t)$ or $q \in \partial P^*$. The latter is impossible as $q \in \partial C$ otherwise.

Assume to the contrary that p lies in $int(\mathcal{F}(t))$. If p lies on the boundary of $C \oplus t$, then $q \in \partial C$ which is impossible. Suppose that $p \in int(C \oplus t \cap \mathcal{F}(t))$. For each cell $C \in \mathcal{C}_b$, $C \oplus t$ lies within $b \oplus 28t$. Thus, p lies in the interior of a connected component of $C \oplus t \cap \mathcal{F}(t)$ that is a subset of K. This implies that $p \in int(K)$ and we can perturb p to another $p' \in K$ such that $d(p',t)$ contains a small neighborhood of q. Therefore, q and a small neighborhood of it should have been milled which contradicts that $q \in \partial(M \cap C) \setminus \partial C$. $\square$

Next we establish a bound on the length of the boundary of the milling action. In general, we bound the lengths of the boundaries of $\mathcal{F}(t) \oplus t$ and $\mathcal{F}(t)$ in any region of diameter $O(t)$.

Lemma 4.2 *Let c be a constant. Then the length of the boundary of $\mathcal{F}(t) \oplus t \cap d(p,ct)$ is at most a constant factor times t.*

Proof Overlay a square grid of side length $t/2$ on $d(p,ct)$. By a packing argument, there are at most $(4c+1)^2$ grid squares intersecting $d(p,ct)$. Let x be one such grid square.

The boundary of $\mathcal{F}(t) \oplus t$ in the interior of x consists of disjoint circular arcs $\{\alpha_i\}$, which are either portions of an accessibility arc in $\mathcal{A}(t)$ or an edge of the polygon P^*. (If α_i is a straight line segment, then we take its radius to be infinity.)

For any point $q \in \alpha_i$, define $f(q)$ to be the center of the disk of radius t that touches q and is tangential to α_i. For each circular arc α_i we define a wedge W_i, which consists of line segments $\{\overline{qf(q)} : q \in \alpha_i\}$. We are going to charge the length of α_i to the intersection of W_i and the boundary of x. It is straightforward to see that the length of the intersection of W_i and the boundary of x is no less than a constant times the length of α_i. The proof will be complete if we can show each point on the boundary of x will be charged at most once.

We claim that no two wedges W_i and W_j intersect each other, which implies that each point on the boundary of x is charged at most once. Assume for contradiction, that wedges W_i and W_j intersect. Then clearly there must be points $q_1 \in \alpha_i$ and $q_2 \in \alpha_j$ such that segments $\overline{q_1 f(q_1)}$ and $\overline{q_2 f(q_2)}$ cross each other. By definition of f, $|\overline{q_1 f(q_1)}| = |\overline{q_2 f(q_2)}| = t$. Further, since q_2 lies on the boundary of $\mathcal{F}(t) \oplus t$, it follows that the disk $d(f(q_1), t)$ does not overlap point q_2. Thus $|\overline{q_2 f(q_1)}| \geq t$. Similarly, $|\overline{q_1 f(q_2)}| \geq t$. But this implies that in the quadrilateral $q_1 q_2 f(q_1) f(q_2)$, the sum of the length of the two opposite sides exceeds the sum of the length of the two diagonals or $f(q_1) = f(q_2)$, in which case the segments $\overline{q_1 f(q_1)}$ and $\overline{q_2 f(q_2)}$ do not cross each other. In either case, we obtain the desired contradiction.

□

Lemma 4.3 *Let c be a constant. Then the length of the boundary of $\mathcal{F}(t) \cap d(p, ct)$ is at most a constant factor times t.*

Proof The proof is similar to the one given for the previous lemma. We only mention the main difference, which concerns the definition of the wedges. The boundaries of $\mathcal{F}(t)$ in the interior of x consists of disjoint circular arcs $\{\alpha_i\}$. For any point $q \in \alpha_i$, define $f(q)$ to be the point on the boundary of the polygon that touches the disk of radius t with center at q. For each circular arc α_i we define a wedge W_i, which consists of line segments $\{\overline{qf(q)} : q \in \alpha_i\}$. We omit the rest of the argument, which is analogous.

□

Lemma 4.4 *The cost of each large-tool BMA for t is $O(t)$.*

Proof Placing t with center on ∂K costs $2t$. Then we move t along the boundary of K and then zig-zag inside K along vertical segments that are separated by a distance at least $2t$ apart. By the result in [AFM95], the cost of this is proportional to the sum of the boundary length of K and the total length of the vertical segments. Since the total length of the vertical segments is $O(t)$ as $K \subseteq b \oplus 28t$, and by Lemma 4.3 the boundary length of K is $O(t)$, the cost of each *large_tool*(t, K, M) is $O(t)$. □

Next we show that the number of large-tool BMAs is $O(mN)$. In order to prove this, we will first need to establish a technical lemma.

Lemma 4.5 *If t is a tool such that $\mathcal{F}(t)$ is connected, then for any box b of side length t, the number of connected components of $b \oplus 2t \cap \mathcal{F}(t)$ that intersect b is $O(1)$.*

Proof Let D denote any set of placements of tool t whose centers lie in b such that each placement lies in a different connected component of $b \oplus 2t \cap \mathcal{F}(t)$. Clearly any upper bound on the size of D is an upper bound on the desired number of connected components. Let $t' = ct$ for some positive real parameter $c < 1$, to be specified later. Let D' denote a set of disks with the same centers as the disks of D, but whose radii are t'. We will show that no three disks in D' have pairwise nonempty intersection. From this, and a simple packing argument, it follows that the number of disks in D', and hence the number of disks in D is $O(1)$. The key idea is that since two disks in D and hence D' are centered in different connected components, there is a bottleneck segment lying between their centers. The distance of the segment from the centers decreases as c becomes small. If three disks in D' have pairwise nonempty intersections, then the center of the middle disk will be sandwiched between two bottleneck segments. Thus, if c becomes small, two bottleneck segments define a thin quadrilateral such that the middle disk is too big to cross the two extremely short sides. The middle disk cannot cross the two longer sides as they are bottleneck segments. Thus, the middle disk cannot move freely around in P^* which contradicts that $\mathcal{F}(t)$ is connected.

Consider the Voronoi diagram of the boundary of P. Recall the bottleneck segments described earlier. Consider the subset of bottleneck segments whose lengths are less than $2t$. It follows from standard results on Delaunay triangulations and Voronoi diagrams that these segments have pairwise disjoint interiors (but they may share a common endpoint).

We begin by showing that if two disks of D' overlap, then there is a bottleneck segment that intersects the line segment joining their two centers. Consider two placements of t from D, such that the disks of radius t' with the same centers overlap each other. Let p and q denote their centers. (See Fig. 6(a).) Without loss of generality, assume that pq is horizontal. Let R be a rectangle with height $2t$ and with p and q at the midpoints of the two vertical sides. Because p and q are in D, they are in different connected components of $b \oplus 2t \cap \mathcal{F}(t)$. This means that it is not possible to move a tool t with center from p to q without moving the center outside R. We interpret this differently. We shrink the disks centered at p and q to a single point. Simultaneously, we expand each point on the boundary of $R \oplus t$ and each point on the domain boundary inside $R \oplus t$ to a disk of radius t. Now, it is not possible to translate tool t from p to q if and only if some expanded disks together separate p from q in $R \oplus t$. So there are two overlapping expanded disks such that the line segment connecting their centers intersect pq. Thus, there is a pair of points on the domain boundary at distance less than $2t$ apart and the line segment connecting them intersects pq. Pick the closest pairs among the above pairs of points. Then among the closest pairs, pick the pair x, y such that the connecting line segment xy has the leftmost intersection with pq. We claim that xy is a bottleneck segment. xy does not cross pq at p or q, otherwise either x or y would like inside the disk of D centered at p or q. Let C be the diametrical circle of xy. The contact point x lies on a domain edge s which is either a horizontal line segment or a circular arc that touches C at x. The same is true for y. Slide a copy C' of C along s to the left by an arbitrarily small distance $\epsilon > 0$ on s. If C' does not encounter any other domain boundary point not on s for some ϵ, then after the sliding, we can expand C' slightly while maintaining emptiness. This shows that xy is a bottleneck segment. Suppose that for all $\epsilon > 0$, C' will encounter some domain

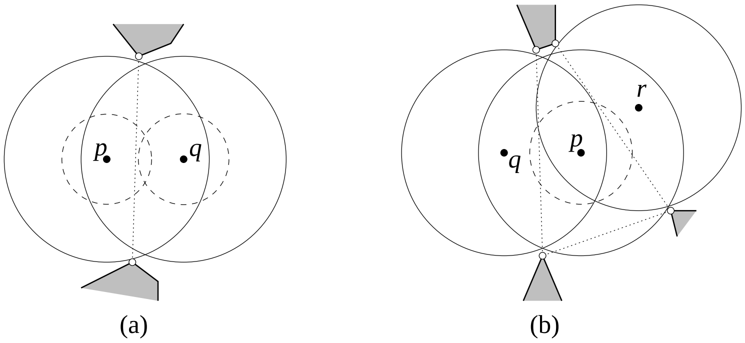

Figure 6: Bounding the number of connected components. The solid circles have radius t, dashed circles have radius t', and dotted lines are bottleneck segments.

boundary point not on s. The first possibility is that C touches another domain boundary point z on the semi-circle to the left of xy, but this means that xz or yz is shorter than xy, contradiction. The second and last possibility is that during the sliding, C' always touches or contains some point on the domain edge containing y. But this contradicts either the shortestness or the leftmostness of xy. Hence, we conclude that xy is a bottleneck segment intersecting pq.

Now, suppose to the contrary that three disks of D' centered at some points p, q, and r have pairwise nonempty intersection. We will show that for all sufficiently small values of c, this will imply that $\mathcal{F}(t)$ is not connected, leading to a contradiction. (See Fig. 6(b).) From the observations of the previous paragraph, it follows that there are bottleneck segments that intersect each edge of the triangle pqr. Hence, there is a point, say p, that lies between the bottleneck segments s_1 and s_2 that intersect pq and pr. Also, s_1 and s_2 intersect a disk of radius $2t'$ centered at p.

Consider the quadrilateral defined by the endpoints of s_1 and s_2 (which may degenerate to a triangle if s_1 and s_2 share a common endpoint). Observe that as the parameter c decreases, s_1 and s_2 intersect a shrinking disk centered at p with p lying between them. Since the lengths of s_1 and s_2 are less than $2t$, and s_1 and s_2 pass arbitrarily close to the center of p, their lengths approach $2t$ as c approaches zero. Since s_1 and s_2 do not intersect, the other two sides of the quadrilateral will fall below any given threshold for sufficiently small c. At the same time, p will lie inside the quadrilateral.

Thus there exists a constant value of c so that all four sides of the quadrilateral are of length less than $2t$. Since p lies inside the quadrilateral whose vertices are points of P's boundary and whose sides are shorter than $2t$, the placement of t at p is effectively trapped at this location. In particular, the connected component of $\mathcal{F}(t)$ containing p has a diameter less than t, and hence lies entirely within $b \oplus 2t$. However, because q and r are in different connected components of $b \oplus 2t \cap \mathcal{F}(t)$, they are in different connected components of $\mathcal{F}(t)$. This contradicts the hypothesis that $\mathcal{F}(t)$ is connected. $\square$

Lemma 4.6 *There are $O(1)$ large-tool BMAs for t induced by any grid square $b \in \mathcal{B}_t$.*

Proof If t is the smallest tool, let $\mathcal{K}$ be the set of connected components of $b \oplus 28t \cap \mathcal{F}(t)$ that overlap $b \oplus 4t$. Otherwise let $\mathcal{K}$ be the set of connected components of $b \oplus 28t \cap \mathcal{F}(t)$ that contain some component of $b \cap \mathcal{F}(2t)$. Since one large-tool BMA for tool t is defined for each component in $\mathcal{K}$, it suffices to show that the number of components in $\mathcal{K}$ is $O(1)$.

Consider the case when t is not the smallest tool. Let K be any component in $\mathcal{K}$. Since K contains a component of $b \cap \mathcal{F}(2t)$, there is a free placement $d(p, 2t)$ where $p \in K \cap b$. Clearly $d(p,t) \subseteq \mathcal{F}(t)$ since a disk of radius t with center in $d(p,t)$ lies completely within $d(p, 2t)$. Further, since $p \in b$, $d(p,t) \subseteq b \oplus 28t$. It follows that $d(p,t) \subseteq K$. By a packing argument, the number of components in $\mathcal{K}$ is $O(1)$.

If t is the smallest tool, then by our assumption $\mathcal{F}(t)$ consists of one connected component. Since b is of side length t, we can cover $b \oplus 28t$ with a constant number of boxes of side length t. Let b' be any such box. Since $(b' \oplus 2t) \subset (b \oplus 28t)$, the number of connected components of $b \oplus 28t \cap \mathcal{F}(t)$ that overlap b' is at most the number of connected components of $b' \oplus 2t \cap \mathcal{F}(t)$ that overlap b' (since expanding the region can only improve connectivity). Thus, by applying Lemma 4.5 to b', it follows that this number of connected components is $O(1)$. This implies that the number of components in $\mathcal{K}$ that overlap $b \oplus 4t$ is $O(1)$.

$\square$

Corollary 4.1 *There are $O(mN)$ large-tool BMAs for all tools.*

Proof For each tool t, we define large-tool BMAs for each $b \in \mathcal{B}_t$. Since a quadtree box of width less than $16t$ can overlap $b \oplus 4t$ for at most a constant number of grid squares b in $\mathcal{G}_t$, the number of squares in $\mathcal{B}_t$ is $O(N)$. By Lemma 4.6, the number of large-tool BMAs induced by each $b \in \mathcal{B}_t$ is $O(1)$. It follows that there are $O(N)$ large-tool BMAs defined for a given tool t. Summing over all tools, we have the desired result. $\square$

4.2 Small-tool basic milling action

Unlike the large-tool BMAs, small-tool BMAs only act on a single cell of the subdivision. At some stage when a small tool first acts on a cell, other tools may have already milled portions of this cell, leaving one or more unmilled regions. We do not know what these tools are, but (as we have already seen with the large-tool BMAs) we design each BMA so that it either mills the entire region or it mills up to an accessibility arc. Henceforth, the term *unmilled component* will refer to an unmilled component that could have resulted by any sequence of BMAs. (Later we will show that no matter what combination of tools have acted on this cell, the number of possible unmilled components that could result is polynomially bounded.) Intuitively, the task of each small-tool BMA is to mill as much material as it can access within an unmilled component such that the tool is always in contact with the unmilled component.

Let U be an unmilled component of C such that t is small for C. Let K be a connected component of $U \oplus t \cap \mathcal{F}(t)$. Define M to be $K \oplus t \cap U$. K and

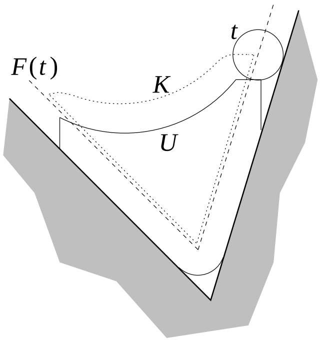

Figure 7: Small-tool BMA. U is bounded by the domain boundary and solid lines. $\mathcal{F}(t)$ is shown with dashed lines, and K is shown with dotted lines. All of U is milled except the portion below the accessibility arc at the bottom.

M define a small-tool BMA by t, denoted $small\_tool(t, K, M)$. This action will move t on the surface with center in K to mill points in M.

As in the large-tool case, we will prove that the unmilled components remaining after a small-tool BMA will be bounded by the boundary of the cell and portions of accessibility arcs. In fact, we will show that for small-tool BMAs, each unmilled component is bounded by at most one accessibility arc. From this we will show that each unmilled component has constant combinatorial complexity.

Lemma 4.7 *After $small\_tool(t, K, M)$ acts on an unmilled component U in a cell C, any resulting unmilled component is bounded by a portion of at least one accessibility arc of radius t.*

Proof Let W be a resulting unmilled component. Take a point q in $\partial W \setminus \partial U$. Such a point q must exist, otherwise U was not affected by $small\_tool(t, K, M)$. By our choice, q is milled by $small\_tool(t, K, M)$ and there is a free placement $d(p, t)$ that touches q, where $p \in K$. We claim that p must lie on the boundary of $\mathcal{F}(t)$ and so $q \in \mathcal{A}(t)$ or $q \in \partial P^*$. The latter is impossible as $q \in \partial C$ otherwise.

Assume to the contrary that p lies in $int(\mathcal{F}(t))$. If p lies on the boundary of $U \oplus t$, then $q \in \partial U$ which is impossible. The other possibility is that p lies in the interior of a connected component of $U \oplus t \cap \mathcal{F}(t)$ which is a subset of K. This implies that $p \in int(K)$ and we can perturb p to another $p' \in K$ such that $d(p', t)$ contains a small neighborhood of q. Thus a small neighborhood of q should have been milled which contradicts that $q \in \partial W \setminus \partial U$. □

Next, we strengthen our result and show that each resulting unmilled component is bounded by exactly one accessibility arc of $\mathcal{A}(t)$. To this end, we need two technical results: Lemma 4.8 and Corollary 4.2. They are illustrated in Fig. 8. Let us think of each edge of P^* as being an open curve (line segment or circular arc). Observe that when an accessibility arc of some tool t is incident to an edge of P^*, the point of incidence subdivides this edge into two portions. Locally about the point of incidence, one portion contains points that are accessible to t and the other contains points that are not. Points that are not accessible to t are said to lie *outside* the accessibility arc. Observe that unmilled

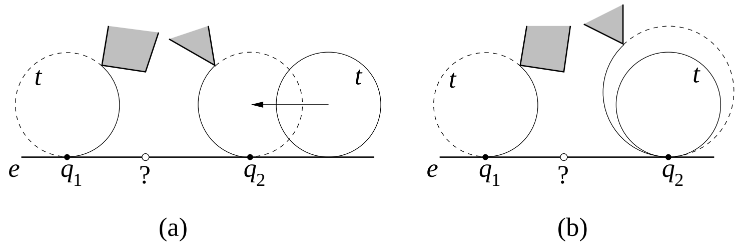

Figure 8: Facing accessibility arcs.

regions are always locally outside of any accessibility arcs on their boundaries. Two accessibility arcs that are incident to the same edge are said to *face* each other if the points between these accessibility arcs lie outside of both arcs. We show that, because of the introduction of bottleneck points, it is not possible for two facing accessibility arcs to be incident to the same edge of P^*.

Lemma 4.8 *If the interior of an edge of P^* is incident to an accessibility arc for tool t, then no point on the outside portion of the edge is accessible to t.*

Proof Assume to the contrary that a domain edge e is incident at a point q_1 to an accessibility arc of radius t and that there a free placement of t that intersects a point of e that is outside this arc. (See Fig. 8(a).) Consider the point of contact q_2 of such a placement that is closest to q_1. Clearly the placement must be tangential to e at this point. Since the placement cannot be moved closer to q_1, there must be a second accessibility arc incident to e at q_2 such that both accessibility arcs face each other. Because these accessibility arcs are blocked by some other boundary points, it follows that their centers lie on the Voronoi diagram of P^*. The Voronoi distance function $v(p)$ is equal to these t at each center and is smaller in between (for otherwise there would be a free placement of t that is closer to q_1 along e). Therefore, there must be a bottleneck point somewhere within the segment q_1q_2, contradicting the hypothesis that they both lie on the same edge of P^*. □

Corollary 4.2 *An edge of P^* cannot be incident to two accessibility arcs (of possibly different sized tools) that face one another.*

Proof If two such arcs exist, then the arc with a larger radius can accommodate a disk that has the same radius as the other arc and touches the domain edge. (See Fig. 8(b).) This contradicts Lemma 4.8. □

Lemma 4.9 *After the action small_tool(t, K, M), each resulting unmilled component W is bounded by points belonging either to ∂C or to exactly one accessibility arc of $\mathcal{A}(t)$.*

Proof By Lemma 4.7, ∂W contains a portion of some accessibility arcs of radius t. Since t is small for C, the contact points between an accessibility arc, α, and P^* are within $buf(C)$. Recall that there can be at most two edges of P^* within $buf(C)$.

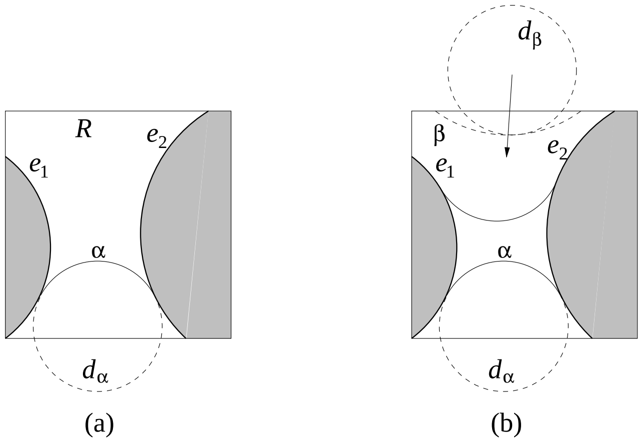

Figure 9: Bounding the combinatorial complexity of unmilled regions for small-tool BMAs.

First we observe that both endpoints of α cannot lie on the same domain edge. This is a simple consequence of the facts that α is a circular arc subtending an angle less than π, the domain edges are either straight line segments or circular arcs, and that $buf(C)$ intersects at most two edges of P^*. Let e_1 and e_2 denote the two domain edges to which α is incident. We assert that each edge is tangentially incident to α. If not, then one endpoint of α must coincide with a vertex of P^*, and the other with one of the edges e_1 or e_2. However, either this vertex is incident to a third edge (contradicting the fact that $buf(C)$ can intersect at most two edges) or else both endpoints of α are incident to a single edge (contradicting the previous observation).

Consider the subregion R of $box(C)$ bounded by α and e_1 and e_2 (See Fig. 9(a)). W lies within R. Suppose that W was bounded by some other accessibility arc β. Since unmilled regions lie outside of their accessibility boundaries, α and β face one another. If β has radius no greater than α's, then β was also produced by a small milling action. By applying the above analysis it follows that β is incident to e_1 and e_2. However, the existence of an edge incident to two accessibility arcs that face one another contradicts Corollary 4.2. Otherwise, if β's radius is greater than α's, then there is a free placement of a disk d_β of radius t that intersects R and lies on the inside of β. (See Fig. 9(b).) If we move d_β towards the disk d_α the fact that α is an accessibility arcs implies that d_β must contact the domain boundary at some point. This contact must be with either e_1 or e_2, by the same analysis used above for α. However, this free placement along such an edge contradicts Lemma 4.8. Hence, we conclude that W is bounded by only one accessibility arc of radius t.

□

4.3 Complexity of unmilled region

In this section, we establish that after any sequence of BMAs the combinatorial complexity of the unmilled region inside a cell is always bounded by a constant. In the sequence, tools can change and large-tool and small-tool BMAs can interleave. The consequence is that within each cell, the number of unmilled components is always bounded by a constant and each unmilled component has constant complexity.

Lemma 4.10 *After any sequence of BMAs on a cell C, the unmilled region in C has constant combinatorial complexity.*

Proof The proof involves two cases, depending on whether the unmilled component resulted from a small-tool or large-tool basic milling action.

Small-tool Case. By Lemma 4.9, after a small-tool BMA, each unmilled component of C is bounded by straight line segments and circular arcs, limited to the four sides of $box(C)$, at most two domain edges of P^*, and at most one accessibility arc. Therefore, each unmilled component has constant combinatorial complexity. Thus, it suffices to show that the number of connected components is bounded by a constant.

Moreover, we assert that each unmilled component U either borders a domain edge intersecting C or a vertex of $box(C)$. To prove this, observe that if U is not bounded by ∂P^*, then for any maximal connected component s of a side of $box(C)$ bounding U, s lies outside of at most one accessibility arc by Lemma 4.9. Thus, the other endpoint of s must be incident to a vertex of $box(C)$. Since there are only a constant number of box vertices, it suffices to show that the number of connected components bounded by domain edges is bounded by a constant.

Each domain edge intersecting C cannot border more than two unmilled components, for otherwise the domain edge would be incident to two accessibility arcs that face each other, contradicting Corollary 4.2. (The worst case occurs when both of the edge's vertices lie within unmilled components, and hence each faces an accessibility arc.) Therefore, the total complexity of all unmilled components produced by small-tool BMAs acting on C is a constant.

Large-tool Case. Let U denote the set of unmilled components of C that resulted from large-tool milling actions. There are two key ideas. First, since we are concerned with large-tool milling actions, the radius of accessibility arcs bounding C is not small compared with $size(C)$. Second, if there are many accessibility arcs bounding U, it implies that there are many placements of tools around U whose radii are not small compared with $size(C)$. This will introduce overlapping among these placements and the overlapping increases as the number of placements increases. For one such tool placement d, the other tool placements overlapping d create an empty region around d which implies that d can be moved a bit into U and mill more. This is impossible. The details are as follows.

We assert that no component of U can be bounded by the accessibility arc of a tool smaller than $size(C)/16$. This is because such an arc would have resulted from the milling action of a small-tool BMA. By definition such a milling action would remove everything reachable to this tool within the component. This implies that no larger tool could later introduce an accessibility arc into the remaining unmilled component. Thus, it suffices to bound the number of accessibility arcs of radius at least $size(C)/16$.

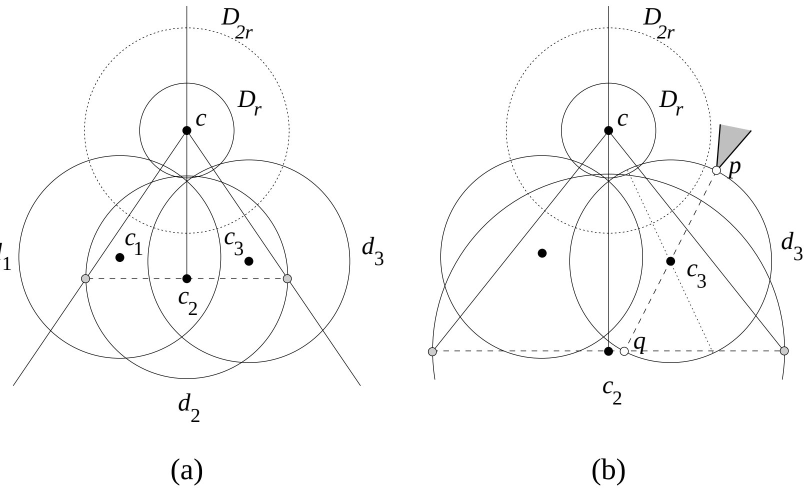

Figure 10: Bounding the combinatorial complexity of unmilled regions for large-tool BMAs.

To simplify the analysis, we overlay a square grid on C of side length $\sqrt{2}r$, where $r = size(C)/32$. The number of such boxes is bounded by a constant. Consider one such box x. Observe that if we enclose x within a disk D_r of radius r, then the closest point outside $buf(C)$ is at distance greater than r from D_r. We will show that the number of accessibility arcs for all large tools t that may contribute to the intersection of ∂U with D_r is $O(1)$. It will follow that the number of accessibility arcs that bound U is also $O(1)$.

Let c be the center of D_r. Let D_{2r} be a disk of radius $2r$ centered at c. Since D_{2r} lies entirely within $buf(C)$, at most two edges of P^* may intersect this disk. (See Fig. 10(a).) Let $\{d_i\}$ be the set of free placements of tools for C that support accessibility arcs that contribute to the intersection of ∂U with D_r. Let $\{c_i\}$ be their respective centers. The radius of each disk is at least $size(C)/16 \geq 2r$. Sort these disks in angular order about c according to locations of their centers. If there are no three consecutive disks d_1, d_2, and d_3 such that $\angle c_1 c c_2 < \pi/6$ and $\angle c_2 c c_3 < \pi/6$, then it follows that there are at most 24 such disks (two per sector of $\pi/6$). We will show that if we exceed this number by more than a small additive constant, then there will be a triple of consecutive disks, d_1, d_2, and d_3, satisfying this condition, such that one of these three disks is free to move further into D_r. However, this will imply that it could not contribute an accessibility arc, a contradiction.

First, by simple trigonometry, any disk d_i of radius at least $2r$ that intersects D_r must intersect D_{2r} along an arc of angle at least $2\arccos(3/4) \approx 1.445 > \pi/3$. Second, if we draw a diameter of d_i perpendicular to cc_i (shown as a dashed line between shaded points in Fig. 10(a)), and join c to one diameter endpoint and c_i, then the angle between these rays is at least $\arctan(2/3) \approx 0.588 > \pi/6$. (In both cases, the minimum occurs when d_i has radius exactly $2r$, and c_i is at distance $3r$ from c.) Third, the center of no d_i can lie inside D_{2r}, otherwise, d_i would completely enclose D_r, implying that d_i contributes no accessibility arc that intersects D_r.

We consider two possible configurations of d_1, d_2, and d_3 depending on the position of the endpoints of the diameter of d_2 perpendicular to cc_2. Call this diameter l_2 (the dashed line segment in Fig. 10(a)). In the first case, both the endpoints of l_2 lie inside d_1 and d_3. By the lower bound on the radii of d_1 and d_3 and the upper bound on the angle their centers subtend about c, it follows that l_2 is fully contained within the union of d_1 and d_3. Since these are both free placements, and since d_2 contributes an accessibility arc, it follows that d_2 must contact ∂P^* along the portion of the arc of d_2 that lies within $D_{2r} \setminus (d_1 \cup d_3)$. By the same reasoning used in Lemma 4.9, it follows that this accessibility arc must contact both of the domain edges of P^* that lie within D_{2r}. However, observe that d_2 is unique in this, since no other accessibility arc can have its contact points lying on the edges of P^* within D_{2r} without violating Corollary 4.2. Thus excluding d_2, no other accessibility arc can be in this configuration.

For the second configuration, either the left endpoint of l_2 lies outside d_1 or the right endpoint of l_2 outside d_3. Let us consider the latter, as the other case is symmetrical. (See Fig. 10(b).) Let l_3 be the diameter of d_3 that lies on the line through c and c_3 (shown as a dotted line in the figure). From the trigonometric observations above it follows that l_3 lies within $D_{2r} \cup d_2$. If the portion of ∂d_3 that contributes the accessibility arc intersects the domain boundary within D_{2r}, then this contact involves one of the two edges of ∂P^* that lies within D_{2r}. Again, by Corollary 4.2, this can only happen for a constant number of accessibility arcs. Otherwise, the closest contact of d_3 with the domain boundary occurs at some point p that lies outside D_{2r}. The point q on ∂d_3 that is diametrically opposite to p, lies in d_2. However, the arc of length π from q to p is free from contact with the domain boundary, contradicting the hypothesis that d_3 contributes an accessibility arc. □

Lemma 4.11 *Given an unmilled component U in a cell C and a tool t small for C, $U \oplus t \cap \mathcal{F}(t)$ consists of a constant number of components, each of constant combinatorial complexity.*

Proof Let x be $\frac{3}{2} box(C)$. If t moves with center in x, then t is entirely inside $buf(C)$. Since there are at most two domain edges intersecting $buf(C)$, the complexity of $x \cap \mathcal{F}(t)$ is bounded by a constant. By Lemma 4.10, the complexity of U is bounded by a constant. Since $U \subseteq C$ and t is small for C, $U \oplus t \subseteq x$. So $U \oplus t \cap \mathcal{F}(t) = U \oplus t \cap (x \cap \mathcal{F}(t))$ which is the intersection of two shapes of constant combinatorial complexities. Thus, we conclude that $U \oplus t \cap \mathcal{F}(t)$ consists of a constant number of components, each of constant combinatorial complexity. □

Lemma 4.12 *There are $O(N(mn)^{O(1)})$ basic milling actions by small tools.*

Proof It suffices to prove that there are $O((mn)^{O(1)})$ basic milling actions by small tools in a cell C. By Lemma 4.1, Lemma 4.9, and Lemma 4.10, after any sequence of BMAs the boundary of the unmilled region in C is bounded by at most a constant c elements of the following varieties: line segments on the boundary of $box(C)$, the at most two domain edges intersecting $box(C)$, and accessibility arcs. There are m different tools and there are $O(n)$ accessibility arcs for each tool size. Therefore, there are $O(m^c n^c)$ possible unmilled regions which may be generated after some sequence of milling actions. Hence, there are $O(m^c n^c)$ unmilled components that a basic milling action by a small tool t (with respect to C) may act on. Given an unmilled component U, $U \oplus t \cap \mathcal{F}(t)$ consists of a constant number of components by Lemma 4.11. This gives rise to a constant number of basic milling actions by t on U. In all, there are at most $O(m^{c+1}n^c)$ basic milling actions by some small tool in C. Summing over all cells, we obtain the bound $O(m^{c+1}n^c N)$. □

5 Approximating optimal milling using BMAs

The main result of this section is to show that any milling plan can be converted into a milling plan consisting entirely of BMAs while sacrificing at most a constant factor in cost.

We associate with each BMA *large_tool*(t, K, M) or *small_tool*(t, K, M) a *starting point*, which may be any point in K. When we perform a BMA, the tool will first be placed at the starting point and at the end, the tool is returned to this point.

Define *mill*(t, K, M) to be the milling cost of the milling operation defined by the BMA *large_tool*(t, K, M) or *small_tool*(t, K, M). Given a set of BMAs S, define S_t to be the subset of S which uses tool t. Define $TSP(S)$ to be the length of a minimum Euclidean traveling salesman tour on the starting points of S and the tool-change center. The cost of S is composed of two elements: the time required to perform each of its milling operations and the time to move from the starting point of one to the starting point of another. Define *mill*(S) to be the sum of *mill*(t, K, M) for all *large_tool*(t, K, M) and *small_tool*(t, K, M) in S. Define $\mathit{transport}(S) = \sum_t TSP(S_t)$.

We immediately have the following.

Lemma 5.1 *Let S be a set of BMAs that mills P^*. Then there exists a milling plan Ψ, using tools in S, such that*

$$\mathit{mill}(\Psi) = \mathit{mill}(S), \qquad \mathit{transport}(\Psi) = \mathit{transport}(S).$$

Conversely we assert that any milling plan Ψ can be transformed into a set of BMAs that mills P^*, whose milling and moving times are comparable to those of Ψ. Before proving this main result, we prove three technical lemmas.

Lemma 5.2 *Consider any path of length L among a uniform rectangular grid of side length s, and let i be the number of cells of the grid that the path intersects. Then $i \le 2\sqrt{2}L/s + 4$.*

Proof Replace the path by a rectilinear path by breaking it at its intersection points with the grid. The resulting path is longer by a factor of at most $\sqrt{2}$. In the worst case, the path starts very close to a vertex, and so it can visit four cells within an arbitrarily small distance. After this, with each walk along the path by distance s the path can visit at most two new cells. Thus the number of cells visited satisfies $i - 4 \le 2\sqrt{2}L/s$. □

Lemma 5.3 *For any path ρ of length L, there is a sequence of disks $d(p_i, 2t)$, where $1 \leq i \leq \lceil L/t \rceil$ and p_i lies on ρ, such that $\rho \oplus t \subseteq \bigcup_i d(p_i, 2t)$.*

Proof Put $d(p_1, 2t)$ at an endpoint p_1 of ρ. Traverse ρ from p_1 to the other endpoint. When ρ leaves $d(p_1, t)$ for the first time, put $d(p_2, 2t)$ at the exit point. Repeat the above until we reach the other endpoint of ρ. The result follows by observing that each placement is separated by an arc length of at least t. $\square$

Lemma 5.4 *Let C_1 and C_2 be two cells such that $size(C_1) \geq size(C_2)$. If $(box(C_1) \oplus t) \cap (box(C_2) \oplus t)$ is nonempty for some $t \leq size(C_1)/3$, then $size(C_2) \geq size(C_1)/100$. Hence, for any point q, there are $O(1)$ cells C such that $box(C) \oplus (size(C)/3)$ contains q.*

Proof Assume to the contrary that $size(C_2) < size(C_1)/100$. Since $t \leq size(C_1)/3$ and $(box(C_1) \oplus t) \cap (box(C_2) \oplus t)$ is nonempty, both the horizontal and the vertical distances between the centers of $box(C_1)$ and $box(C_2)$ is less than $7size(C_1)/6 + size(C_2)/2 < 1.4size(C_1)$. The buffer zone of the parent of $box(C_2)$ lies inside $9box(C_2)$ and has width at most $0.09size(C_1)$. Thus, the buffer zone of the parent of $box(C_2)$ lies inside $3box(C_1)$ which is the buffer zone of $box(C_1)$. This implies that the buffer zone of the parent of $box(C_2)$ intersects at most two domain edges, which contradicts the splitting of it. Therefore, we conclude that $size(C_2) \geq size(C_1)/100$.

Take any point q. Let C be the cell of the largest size such that $q \in box(C) \oplus (size(C)/3)$. Thus, a square of width $3size(C)$ centered at q contains all the boxes $box(C')$ for some cells C' such that $q \in box(C') \oplus (size(C')/3)$. From the above $size(C') = \Theta(size(C))$. So a packing argument shows that there are $O(1)$ of such boxes. Since each cell is a connected component of a box and each box intersects at most a constant number of domain edges, the number of cells is also $O(1)$. $\square$

Theorem 5.1 *Let Ψ be a milling plan. Then there exists a set S of BMAs which mills P^* using at most twice the number of tools as in Ψ, such that $mill(S) = O(mill(\Psi))$ and $transport(S) = O(mill(\Psi) + transport(\Psi))$.*

The proof of this theorem is presented in the remainder of this section. We first identify a set S_1 of large-tool BMAs and then a set S_2 of small-tool BMAs that mills P^*. Clearly, $mill(S_1 \cup S_2) = mill(S_1) + mill(S_2)$ and $transport(S_1 \cup S_2) \leq transport(S_1) + transport(S_2)$. Thus, it suffices to bound the milling and transport costs of S_1 and S_2 separately by $mill(\Psi)$ and $transport(\Psi)$. Each of S_1 and S_2 will involve at most the same number of tools used in Ψ. Thus $ntools(S_1 \cup S_2) \leq 2ntools(\Psi)$. For each tool t, Ψ_t denotes the set of milling paths involving t, and $\Psi_{\leq t}$ denotes the set of milling paths involving t or smaller tools. We think of Ψ_t as the set of paths along which the center of the tool moves and the same applies for $\Psi_{\leq t}$. To complete the proof, we present the analyses of the large-tool and small-tool cases in the next two subsections.

5.1 Large-Tool BMAs

For each continuous curve η_t in Ψ_t, we find a set of large-tool BMAs so that if a point q in a cell of size less than $4t$ is milled by t traversing along η_t, then q is milled by some large-tool BMA in this set. Let t' be the smallest tool in the range $(t/4, t]$. Let $X_{t'}$ be the squares b in the grid $\mathcal{G}_{t'}$ through which η_t passes. For each square $b \in X_{t'}$, we add to S_1 all large-tool BMAs $large\_tool(t', K, M)$ induced by b such that K contains a point on η_t inside b. We claim the following:

(1) If a point q in some cell C is milled by a tool in Ψ of size greater than $size(C)/4$, then q is milled by a large-tool BMA in S_1.

(2) $mill(S_1) = O(mill(\Psi))$ and $transport(S_1) = O(mill(\Psi) + transport(\Psi))$.

To prove (1), since q is milled by some tool t of size greater than $size(C)/4$, $q \in d(p,t)$ for some point p on a curve η_t in Ψ_t. Let b be the square in $\mathcal{G}_{t'}$ that contains p. Let K be the connected component of $(b \oplus 28t') \cap \mathcal{F}(t')$ that contains p. (Note that K must exist as $b \oplus 28t'$ contains p and $\mathcal{F}(t) \subseteq \mathcal{F}(t')$.) We claim that b induces the BMA $large\_tool(t', K, M)$ which is added to S_1. Since $p \in b$ and $d(p,t)$ overlaps C and $t < 4t'$, this implies that $b \oplus 4t'$ overlaps C. Also, since $size(C) < 4t$ and $t < 4t'$, $size(C) < 16t'$. If t' is the smallest tool, then we are done. Otherwise, we need to check if K contains a component of $b \cap \mathcal{F}(2t')$. Since $t \geq 2t'$ by choice of t', $\mathcal{F}(t) \subseteq \mathcal{F}(2t')$. Since $p \in \mathcal{F}(t)$ and $p \in K$, we conclude that K contains a component of $b \cap \mathcal{F}(2t')$. Finally, we verify that $large\_tool(t', K, M)$ mills q. Recall that $q \in d(p,t)$ for some p on η_t and $p \in K$. Thus, one can first center t' at p and then move within $d(p,t)$ to mill q.

To prove (2), by Lemma 5.2, each η_t in Ψ_t passes through $O(L_{\eta_t}/t + 1)$ squares in $\mathcal{G}_{t'}$ and so $X_{t'}$ has $O(L_{\eta_t}/t + 1)$ squares. By Lemma 4.6, each square in $X_{t'}$ induces $O(1)$ large-tool BMAs for t'. So the total number of large-tool BMAs identified for η_t is $O(L_{\eta_t}/t + 1)$. By Lemma 4.4, the total milling cost of these BMAs is $O(t'(L_{\eta_t}/t) + t') = O(L_{\eta_t} + t')$ which is bounded by the milling cost of η_t (length plus placement cost $2t$). Thus summing over Ψ_t for all t, we have $mill(S_1) = O(mill(\Psi))$.

We can visit all the large-tool BMAs in S_1 involving tool t as follows. Follow Ψ to transport t to a point on η_t. Then transport t to an endpoint of η_t (this costs $O(L_{\eta_t})$. Transport to the starting points of the large-tool BMAs defined at this endpoint and apply the BMAs. Traverse along η_t to the center of the next disk in D_{η_t} and repeat the application of BMAs (this costs $O(L_{\eta_t} + t)$). Finally, transport t to the point on η_t from where Ψ will leave η_t (this costs $O(L_{\eta_t})$). Thus, the entire tour can be viewed as the transport of t in Ψ plus some detour. The cost of the detour sums to $O(mill(\Psi_t))$. Thus, $transport(S_1) = O(mill(\Psi) + transport(\Psi))$.

5.2 Small-Tool BMAs

We assume that all the large-tool BMAs in S_1 have been applied. Let C be a cell with an unmilled region. Any tool t in Ψ that acts on this unmilled region must satisfy $t \leq size(C)/4$ and so t must be small for C. We will identify a set S_2 of small-tool BMAs to mill the rest of P^* and charge the cost to $mill(\Psi)$ and $transport(\Psi)$.

The charging scheme for the small-tool BMAs is more complex than for large tools. Consider some unmilled component. Let t be the largest tool used by Ψ to mill any point of this component. We will introduce the corresponding small-tool BMA for t to mill as much of this component as possible without losing contact with it. The key to establishing the approximation bound is to show that no combination of smaller tools could mill the same region with significantly less cost. Recall from Section 2 and Fig. 1 that one reason that larger tools are not necessarily better than smaller tools is that a large tool may only shave away a small amount of additional material, into which a small tool may be able to plunge deeply. Intuitively, if tool t does plunge deeply into the unmilled region, then it will mill more efficiently than a smaller tool. On the other hand, if t does not plunge deeply into the unmilled region, then the small-tool BMA will scrape along the boundary of the unmilled region. To account for this, we will introduce a charging scheme to pay for this milling action. The boundary of each unmilled region will be assigned a charge proportional to its length. We will then show that the total charges will be dominated by the other costs of our milling plan.

To facilitate the charging, we need to initialize some charge on boundaries of unmilled regions after applying all large-tool BMAs in S_1. For each segment on these boundaries, we associate a charge proportional to its length. We claim that these charges can be paid for by $mill(\Psi)$ and the argument is as follows.

By Lemma 4.2, the sum of lengths of segments that lie on accessibility arcs left by large-tool BMAs in S_1 can already be paid for by $mill(S_1)$, which is $O(mill(\Psi))$. The other boundary segments of the unmilled regions lie either on the boundary of quadtree boxes or domain edges. Let s be a boundary segment of the unmilled region in a cell C that lies either on the boundary of $box(C)$ or on some domain edge bounding C. Let t be the largest tool in Ψ that mills any point on s. Consider $s \oplus t$.

Suppose that s is a straight line segment. Let t' be the tool traversing a subpath $\rho_{t'}$ in $\Psi \cap s \oplus (4t/3)$ such that $\rho_{t'} \oplus t'$ overlaps s. Note that $t' \leq t$. By Lemma 5.3, the perimeter of $\rho_{t'} \oplus t'$ is less than $|\rho_{t'}| + O(t')$. If we take all such tools and subpaths, then s is covered by $\bigcup_{t'} \rho_{t'} \oplus t'$ and so $|s| \leq \sum_{t'} |\rho_{t'}| + O(t')$. Notice that each $\rho_{t'}$ either traverses a distance of at least $t/3$ in $s \oplus (4t/3) \setminus (s \oplus t)$ or contains a placement of t'. Thus, the $O(t')$ term can be charged to this placement or $\rho_{t'}$ itself. Thus, $|s|$ can be charged to the total cost of such milling subpaths $\rho_{t'}$. Hence, by Lemma 5.4 and Lemma 4.10, the total length of straight line segments of the unmilled regions can be charged to $O(mill(\Psi))$. The possibility that s is a circular arc can be handled identically. This proves our claim that charges associated with boundaries of unmilled regions after applying large-tool BMAs in S_1 can be paid for by $O(mill(\Psi))$.

Let $\mathcal{U}$ be the set of unmilled components after applying large-tool BMAs in S_1. We define the set S_2 of small-tool BMAs iteratively as follows. Remove $U \in \mathcal{U}$. Let t be the largest tool in Ψ that mills any point of U. For each connected component K of $(U \oplus t) \cap \mathcal{F}(t)$ that intersects Ψ_t, we add $small\_tool(t, K, M)$ to S_2. Then we subtract $K \oplus t$ from U for each such K and put the unmilled components produced back to $\mathcal{U}$. We repeat the above until $\mathcal{U}$ becomes empty.

We bound $mill(S_2)$ by bounding the placement costs and milling path lengths separately. Take any $small\_tool(t, K, M) \in S_2$. In Ψ_t, there is either a placement of t in $K \oplus t$ or a segment η of length t in $(K \oplus t) \setminus K$. We charge the placement cost of $small\_tool(t, K, M)$ to this placement in $K \oplus t$ or the length

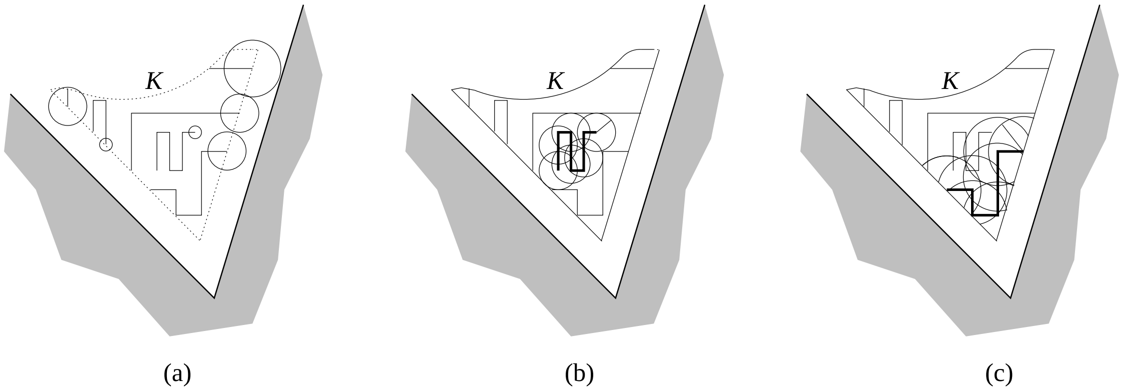

Figure 11: Charging argument for small-tool BMAs. Figure (a) shows $K \cap \Psi_{\leq t}$. Figure (b) shows η^* for one component η shown in bold. Figure (c) shows the trimmed η^* for another component η shown in bold.

of η. At any moment in time, a point q on Ψ_t can only lie inside K for at most a constant number of small-tool BMAs in S_2 by Lemma 5.4, Lemma 4.10, and Lemma 4.11. Moreover, after applying $small\_tool(t, K, M)$, q is at distance at least t away from any new unmilled component produced. Thus, q cannot be charged again for these new unmilled components or any subset of them. Therefore, the placement costs of small-tool BMAs in S_2 is bounded by $O(\sum_t mill(\Psi_t)) = O(mill(\Psi))$.

To bound the milling path length of $small\_tool(t, K, M)$, consider the intersection $\Gamma = K \cap \Psi_{\leq t}$. $\Gamma \cup \partial K$ is an arrangement of curves (possibly consisting of several connected components). (See Fig. 11(a), for example.) For each tool t' used in $\Psi_{\leq t}$, we let $\eta_{t'}$ denote the portion of a milling path for tool t' that lies in Γ. If we move t along ∂K and t' along $\eta_{t'}$ for each $\eta_{t'} \in \Gamma$, we must mill the entire $K \oplus t$. The reason is as follows.

Let q be any point in $K \oplus t$. If q is at distance $\leq t$ from some point on ∂K, then it must be milled as t is moved along ∂K. If q is at distance $> t$ from every point on ∂K (i.e. $q \in K \ominus t$), then q must lie inside U. Further, since t is the largest tool in Ψ that mills any point of U, q must be milled by Ψ using a tool t' of size $\leq t$. Clearly, the center of such a tool t' lies within K.

Our plan is roughly to move t almost along $\Gamma \cup \partial K$ to bound the milling path length of $small\_tool(t, K, M)$. The main problem with this strategy is that if Γ consists of many different components, then the placement cost of t for each component cannot be paid for by the much smaller placement costs that may have been incurred by Ψ. To remedy this, we will add extra segments to connect all components in $\Gamma \cup \partial K$; we will show that the length of these segments can be paid for either by the length of these components or by the placement costs in Ψ.

Each connected component η of $\Gamma \cup \partial K$ can be viewed as a single oriented curve by an Eulerian traversal. Let t_η be the largest tool used in η (we assume that t is used for ∂K). By Lemma 5.3, we can find $L_\eta / t_\eta + 1$ disks of radius $2t_\eta$ to cover $\eta \oplus t_\eta$, where L_η is the length of η. We add a straight line segment of length $2t_\eta$ to connect the envelope of the union of these disks and η. We denote the union of this extra line segment, the boundaries of the disks and

η collectively by η^*. (For example, see Fig. 11(b).) The total length of η^* is $O(L_\eta + t_\eta)$. Also, let η^R denote the region formed by taking the union of these disks. If we could move t_η along η^*, then this would mill all of η^R. However, it may not be possible to move t_η along η^*, if it projects outside of K. Therefore, we trim η^R by taking its intersection with K. Also, we trim η^* by taking its intersection with K and then adding the portion of K that lies within the (trimmed) η^R. (For example, see Fig. 11(c).)

We claim that moving t_η along η^* mills all of η^R. For the sake of contradiction, assume that there is a point q in η^R which cannot be milled by moving t_η along η^*. Let q be inside disk x. Then q cannot be within distance t_η of center of disk x because moving t_η along η would mill q and η^* contains η since η lies entirely within K. Let p denote the closest point to q on the boundary of disk x. Clearly p must lie outside of K, else t_η would be placed at p and it would mill q. Thus, there must be a point on the boundary of K which intersects segment qp and moving t_η along the portion of K included in η^* would mill q, contradiction. This proves our claim that moving t_η along η^* mills all of η^R.

Earlier we proved that moving t along ∂K and t' along $\eta_{t'}$ for each $\eta_{t'} \in \Gamma$, we must mill the entire $K \oplus t$. It follows that $\bigcup_{\eta\in\Gamma\cup\partial K}(\eta \oplus t_\eta) \cap K = K$. Since η^R contains $(\eta \oplus t_\eta) \cap K$, $\bigcup_{\eta\in\Gamma\cup\partial K} \eta^R = K$. Let η^0 be the connected component in the arrangement $\bigcup_{\eta\in\Gamma\cup\partial K} \eta^*$ that contains ∂K. We claim that by moving t along η^0, we mill the entire $K \oplus t$. The argument is as follows.

Suppose there is a point $q \in K \oplus t$, which is not milled by moving t along η^0. Certainly, q cannot be within distance t of some point on ∂K, since all such points are milled as t is moved along ∂K. Assume therefore that q lies within $K \ominus t$. Let η^1 denote the union of all components in the arrangement $\bigcup_{\eta\in\Gamma\cup\partial K} \eta^*$ for which the component contains some η^* such that η^R contains q. Since moving t along any component in η^1 would mill q, it follows that no component in η^1 is connected to ∂K. But this implies that there is a point sufficiently close to the boundary of η^1 (and therefore inside K), which is not contained in $\bigcup_{\eta\in\Gamma\cup\partial K} \eta^R$. This contradicts what we proved earlier and so moving t along η^0 must mill $K \oplus t$.

Next we bound the length of the milling path for applying $\mathit{small\_tool}(t, K, M)$. Clearly this is bounded by the length of η^0. Let ξ be the connected component of $\Gamma \cup \partial K$ that contains ∂K. Then the length of the milling path for applying $\mathit{small\_tool}(t, K, M)$ is bounded by the sum of length of ξ^* and $O(L^*)$, where $L^* = \sum_{\eta\in\Gamma\cup\partial K, \eta\neq\xi} L_\eta + t_\eta$.

L^* is bounded by the sum of lengths of $K \cap \Psi_{\leq t}$ and the associated placement costs of $\Psi_{\leq t}$ in K. At any moment in time, a point q on $\Psi_{\leq t}$ can only lie inside K for at most a constant number of small-tool BMAs $\mathit{small\_tool}(t, K, M)$ by Lemma 5.4, Lemma 4.10, and Lemma 4.11. Hence, q lies on the milling path of only a constant number of small-tool BMAs in S_2 that can be applied at this moment. Moreover, after applying one such $\mathit{small\_tool}(t, K, M)$, the same point q is at distance at least t away from any new unmilled component produced. Thus, q cannot be used again in the future in the milling paths for these new unmilled components or any subset of them. Therefore, summing L^* over all small-tool BMAs in S_2 is $O(\mathit{mill}(\Psi))$. The length of ξ^* is bounded by the sum of length of ∂K plus the length of $K \cap \Psi_{\leq t}$ plus $O(t)$. The $O(t)$ term can be absorbed like the placement cost of $\mathit{small\_tool}(t, K, M)$. As analyzed before, the sum of the lengths of $K \cap \Psi_{\leq t}$ over all small-tool BMAs in S_2 is $O(\mathit{mill}(\Psi))$. It remains to bound the length of ∂K.

First, by Lemma 4.11, there is a constant number of segments in ∂K. Second, we claim that any placement of t on ∂K contains a point on ∂U. Otherwise, such a placement must then lie in the interior of U and its center must lie on $\partial\mathcal{F}(t)$. But this implies that the interior of U contains a point on the boundary of the domain, contradiction. Now, we are ready to bound the length of ∂K as follows. Move the tool t with center along each boundary segment s on ∂K. By our second claim, moving t with center along s will eliminate points on ∂U. Moreover, the length of s is at most the length of segments on ∂U eliminated + $O(t)$. By our first claim, the total length of ∂K is bounded by $O(t)$ plus the length of segments on ∂U eliminated while moving t along ∂K. The $O(t)$ term can be absorbed like the placement cost of $\mathit{small\_tool}(t, K, M)$. ∂U consists of segments of two possible kinds. The first kind consists of segments on boundaries of unmilled regions left after applying large tool BMAs in S_1. By our initial setup, these segments carry enough charge to pay for themselves. The second kind consists of accessibility arcs left after applying some small tool BMAs introduced earlier to S_2. By Lemma 4.9, a small tool BMA will introduce at most a constant number of accessibility arcs. Thus, we can charge the sum of lengths of these accessibility arcs to the placement cost of the small tool BMA introducing them. As analyzed before, the sum of placement costs of small tool BMAs in S_2 is bounded by $O(\mathit{mill}(\Psi))$.

The above establishes that $\mathit{mill}(S_2) = O(\mathit{mill}(\Psi))$. Given any tool t, we transport t as follows. Take out all the small-tool BMAs $\mathit{small\_tool}(t, K, M)$ in S_2 involving t. By construction, Ψ_t visits some point q_K in each K. Hence, we can visit all the small-tool BMAs in S_2 involving t by following Ψ_t which costs $\mathit{mill}(\Psi_t) + \mathit{transport}(\Psi_t)$. At each small-tool BMA $\mathit{small\_tool}(t, K, M)$ visited, we take a detour from q_K to the starting point specified for $\mathit{small\_tool}(t, K, M)$, mill points in M, and finally return to q_K. The trip from q_K to the starting point and the final return to q_K costs no more than the milling path length of $\mathit{small\_tool}(t, K, M)$. Therefore, $\mathit{transport}(S_2)$ is bounded by $\sum_t \mathit{mill}(\Psi_t) + \mathit{transport}(\Psi_t)$ plus the sum of milling path lengths of small-tool BMAs in S_2. The latter sum is $O(\mathit{mill}(\Psi))$ as analyzed before. Hence, $\mathit{transport}(S_2) = O(\mathit{mill}(\Psi) + \mathit{transport}(\Psi))$.

This completes the proof of Theorem 5.1. □

6 Greedy Approximation

We reduce the problem of finding an optimal milling plan to a *weighted set cover problem*: given a set S and a family $\mathcal{Z}$ of some subsets of S where each $X \in \mathcal{Z}$ is associated with weight $w(X)$, the objective is to find $\mathcal{Y} \subseteq \mathcal{Z}$ such that $S = \bigcup_{X \in \mathcal{Y}} X$ and $\sum_{X \in \mathcal{Y}} w(X)$ is minimized. Denote such an instance by $(S, \mathcal{Z})$. We approximate the weighted set cover problem with a greedy algorithm, thus achieving the claimed logarithmic factor approximation. The difficulty is that, as we will see, the instance of our weighted set cover problem is too large to be described explicitly. Therefore, we run an approximate greedy algorithm on a succinct problem description. We will show that this only adds an extra constant factor.

6.1 Reduction

We refine the subdivision of P^* by overlaying the accessibility arcs of $\mathcal{F}(t)$ for all t on it. The set of faces in the refined subdivision is the set S in our weighted set cover instance. Intuitively, we have to cover all the points in P^* by covering all the faces in S. It remains to define $\mathcal{Z}$.

For each *large_tool*(t, K, M) or *small_tool*(t, K, M), M is a subset of faces in S. We call K a *t-locus* and M a *t-region*. A *t-subset* is union of some t-regions and so a t-subset is a set of faces in S. For each t-subset X, we denote by X_M the set of t-regions forming X and we denote by X_K the collection of corresponding t-loci. The weight $w(X)$ of X is the sum of three terms. The first term is the total milling costs of the t-regions in X_M. The second term is the length of the MST connecting the tool-change center and the starting points of the t-loci in X_K. The third term is the sum of placement costs $O(|X_K| \cdot t)$. These terms give the total cost of milling faces in X using the BMAs induced by X_K.

We first include in $\mathcal{Z}$ the collection of all possible t-subsets for each tool t. Then we prune $\mathcal{Z}$ such that for each tool t, $X \in \mathcal{Z}$, and any cell C, if there are *small_tool*(t, K, M) and *small_tool*(t, K', M') where $K, K' \in X_K$, $M, M' \in X_M$, and $M, M' \subseteq C$, then *small_tool*(t, K, M) and *small_tool*(t, K', M') act on unmilled components of the same unmilled region of C. By our proof of Theorem 5.1, solving $(S, \mathcal{Z})$ will return at least a constant factor approximation of the milling problem.

6.2 Approximate greedy algorithm

If $(S, \mathcal{Z})$ is described explicitly, then we can use the following greedy heuristic to obtain an $O(\log m + \log N)$ approximation factor. Initialize the cover $\mathcal{Y}$ to be empty. Compute the t-subset $X \in \mathcal{Z}$ that minimizes the *average weight* defined to be the ratio $w(X)$ divided by the number of faces in X that are currently in S. Then include X in $\mathcal{Y}$ and remove all the faces in S contained in X. Repeat until S becomes empty. It is well known that this procedure produces a set cover whose total weight is at most $\ln |S|$ times the optimal. In our case, $|S| = O((Nmn)^{O(1)})$. Since $N \geq n$, the approximation ratio is $O(\log m + \log N)$. Unfortunately, it is not efficient to describe $(S, \mathcal{Z})$ explicitly, since $|\mathcal{Z}|$ is exponential in the number of faces in S. Instead we only store the set of t-regions for all tools and compute a t-subset of approximately minimum average weight (within a constant factor) by solving a series of instances of a variant of k-TSP. Hence, we still solve $(S, \mathcal{Z})$ within a logarithmic approximation factor.

6.2.1 Strategy and difficulty

We briefly review the k-TSP problem and introduce a variant that will be solved repeatedly as a subproblem. Given l points in the plane, the k-TSP is to find a tour of minimum length that visits any j of the l given points. The k-TSP problem is NP-hard but it can be approximated to within any constant factor in polynomial time [Aro96, Mit96]. These algorithms for approximating the k-TSP problem are based on showing that computing the optimum tour of a particular structure will provide an approximation to the optimum tour. Then the optimum tour of the particular form can be computed using dynamic programming. We generalize the k-TSP problem to find a tour that collects coins at the visited points. Each point p is given a table $table(p)$ and each entry of $table(p)$ is a coins-cost pair which tells the cost of collecting the associated number of coins at p. The j-WTSP is to find a tour that collects j coins so that the tour length plus the sum of the cost of collecting coins at visited points is minimized. This is defined to be the weight of the tour. The dynamic programming paradigm for approximating the k-TSP is powerful enough to approximate the j-WTSP within a constant factor in polynomial time. The geometric component of the k-TSP approximation algorithm is unaffected by this reduction, so the approximation bounds proved in [Aro96, Mit96] hold here as well.

Our strategy is to compute, for each tool t, a t-subset of approximately minimum average weight, and then return the one of the least average weight. Consider computing this t-subset for tool t. We first outline our approach by making the (invalid) assumption that t-regions are disjoint. We will show how to overcome this afterwards.

Given a t-region M, fix a point p in the corresponding t-locus. Let f_p be the number of faces covered by M. Let c_p be the sum of milling cost and placement cost for the BMA by t that mills M. Create the table $table(p)$ which contains only one entry, namely, (f_p, c_p). Repeat this for all other t-regions. This yields a collection of points and their associated tables. Let F_t be the maximum number of faces currently in S that are covered by some t-region. For each j, $1 \leq j \leq F_t$, find the approximate j-WTSP and compute its average weight. Afterwards, select the tour with the minimum average weight and this corresponds to the t-subset with approximately minimum average weight.

Due to the possible overlapping among t-regions, the number of faces covered when visiting a set of points is not simply the sum of numbers of faces covered when visiting each point. Thus, the above strategy needs to be improved to overcome this difficulty. We describe below two transformations to get around this problem and obtain the desired series of instances of j-WTSP.

6.2.2 Collapsing small-tool BMAs

Let $large\_cell(t)$ denote the set of cells for which t is small. Given a cell $C \in large\_cell(t)$, all the t-regions inside C are generated by small-tool BMAs by t. If $box(C) \oplus (size(C)/3)$ does not contain the tool-change center, then we put a representative point $pt(C)$ at the center of $box(C)$ as the common point for all t-regions in C. If $box(C) \oplus (size(C)/3)$ contains the tool-change center, then we put a representative point $pt(C)$ at the tool-change center. (If $box(C)$ contains two cells, then we can put these two points slightly apart at the center of $box(C)$.) Let $\mathcal{M}$ be the milling plan obtained in Theorem 5.1. We modify $\mathcal{M}$ as follows. First, if t is transported to C and M is the first t-region in C visited by t, then we first transport t to $pt(C)$ and then to M to start milling. Afterwards, if t is transported to $pt(C)$ several times, then we transport t to $pt(C)$ exactly once and then mill all the t-regions in C that t should mill before going to another cell. Let $\mathcal{M}'$ denote the modified milling plan.

Lemma 6.1 *The tour length of t in $\mathcal{M}'$ is within a constant factor of the tour length of t in $\mathcal{M}$.*

Proof We will show that detouring via the common point $pt(C)$ of a cell C increases the tour length of t by a constant factor, and the lemma will follow. Thus, we focus on proving that the detour is not expensive. Let $\mathcal{M}^*$ denote the modification of $\mathcal{M}$ with the detour. Let T and T^* be a tour of t in $\mathcal{M}$ and its modified version in $\mathcal{M}^*$ respectively. We first modify T^* as follows. In T^*, if there is an edge e from the starting point of the t-locus K_1 for a t-region in cell C_1 to $pt(C_2)$ and then to the starting point of the t-locus K_2 for a t-region in cell C_2, then we replace e by a path of two edges: from the starting point of K_1 to the starting point of K_2 and then to $pt(C_2)$. Let $\tilde{T}$ denote the modification of T^*. The length of $\tilde{T}$ is no less than the length of T^* by triangle inequality. We call an edge in $\tilde{T}$ between a t-locus for a t-region in a cell C and $pt(C)$ a *detour edge*. By construction, $\tilde{T}$ contains all the edges in T and some detour edges. Let e be a detour edge from a point p to $pt(C)$ for some cell C. If $box(C) \oplus (size(C)/3)$ contains the tool-change center, then $pt(C)$ is the tool-change center. Then we charge e to the length of the tour T. Suppose that $box(C) \oplus (size(C)/3)$ does not contain the tool-change center. In T, after leaving the point p, the tour has to leave $box(C) \oplus (size(C)/3)$ eventually. Let ρ be the path in T starting at p and ending at the boundary of $box(C) \oplus (size(C)/4)$. The length of ρ is $\Omega(size(C))$. We charge the length of e in $\tilde{T}$ to the length of ρ. We bound the total charge on the length of T in the following. T visits the same box a constant number of times because there are at most two cells in a box and there is a constant number of unmilled components in a cell to act on. Therefore, there is a constant number of detour edges in $\tilde{T}$ for any box. By Lemma 5.4, there are $O(1)$ boxes $box(C)$ such that $box(C) \oplus size(C)/3$ contains the tool-change center. So the length of T is charged a constant number of times by these boxes. For the rest of the charging, observe that when we charge the length of a detour edge for a box $box(C)$ to some subpath ρ in T, ρ lies inside $box(C) \oplus size(C)/3$ and each point on ρ receives at most constant units of charge. By Lemma 5.4, there are at most a constant number of boxes $box(C)$ such that $box(C) \oplus size(C)/3$ contains a particular point in T. Thus, the accumulated charge on each point in T is bounded by a constant. This proves that the length of $\tilde{T}$ and hence T^* is within a constant factor of the length of T. $\square$

For each cell $C \in large\_cell(t)$, we associate a table $table(pt(C))$ with $pt(C)$ to reflect all the possible effects of small-tool BMAs by t in C and the corresponding costs. We enumerate all possible unmilled regions in C and for each unmilled region, we enumerate all possible combinations of small-tool BMAs on unmilled components. Note that each unmilled component induces at most a constant number of small-tool BMAs. For each such combination of small-tool BMAs by t, we find out the number of faces covered and the cost (sum of costs of the small-tool BMAs, the length of the minimum spanning tree connecting $pt(C)$ and the t-loci, and the placement cost). The number of combinations to be evaluated is $O((mN)^{O(1)})$ and the table $table(pt(C))$ has $O((mN)^{O(1)})$ entries.

6.2.3 Separating large-tool BMAs

After collapsing small-tool BMAs by a tool t, we have one representative point for each cell $C \in large\_cell(t)$. We now address the overlapping among t-regions of large-tool BMAs. We divide such t-regions into groups so that within a group, no two t-regions overlap. The division is done by coloring an induced graph as follows. Consider the grid squares that induce the large-tool BMAs by t. There is a constant number of large-tool BMAs induced by each grid square. Since each grid square has width $\Omega(t)$, the t-region of a large-tool BMA cannot overlap more than a constant number of grid squares. Thus, the t-region of a large-tool BMA overlaps at most a constant number of t-regions of other large-tool BMAs. This induces a graph of maximum degree bounded by a constant Δ. Such a graph is colorable using at most $\Delta + 1$ colors which yields at most $\Delta + 1$ groups of t-regions. We fix a point p in each t-region M and associate with it a table $table(p)$ of a single entry, namely, the faces covered by M and cost (cost of the corresponding large-tool BMA plus placement cost). Thus, we obtain at most $\Delta + 1$ groups of points each associated with a table. We denote these groups by $G_i(t)$ for $1 \leq i \leq \Delta + 1$. Finally, we add one last group $G_{\Delta+2}(t)$ which contains $pt(C)$ along with $table(pt(C))$ for each cell $C \in large\_cell(t)$ (i.e., $G_{\Delta+2}(t)$ takes care of the small tool BMAs by t discussed in section 6.2.2.)

Lemma 6.2 *There is a milling plan $\mathcal{A}''$ in which each tour of a tool t beginning and ending at the tool-change center visits only points in some $G_i(t)$. Moreover, $\mathcal{A}''$ approximates the optimal milling plan within a constant factor.*

Proof Consider the milling plan $\mathcal{A}'$ constructed in Lemma 6.1. Let T be a tour of some t in $\mathcal{A}'$ beginning and ending at the tool center. We simply duplicate T in all the groups. We repeat this for all other tours of all tools. The cost of the resulting milling plan is at most $\Delta + 2$ times the cost of $\mathcal{A}'$ which is within a constant factor of the optimal. $\square$

6.3 The complete algorithm

By Lemma 6.2, it suffices to run a greedy heuristic to approximate $\mathcal{A}''$. Lemma 6.2 also implies that for each tool t, we need to consider only t-subsets whose t-regions have points in one single group. Thus, the complete greedy algorithm is as follows. Recall that F_t is the number of faces currently in S covered by some t-region.

Theorem 6.1 *An approximate milling plan can be computed in $O((mN)^{O(1)})$ time with cost within $O(\log m + \log N)$ of the optimal.*

Proof By Lemma 6.2, it suffices to approximate $\mathcal{A}''$. Due the nature of tours in $\mathcal{A}''$, the nested for loops clearly return the t-subset of minimum average weight, if the j-WTSP could be solved exactly. The approximation factor would be $O(\ln |S|) = O(\log m + \log N)$ by the well-known performance of greedy heuristic. The j-WTSP is approximated to within a constant factor in time $O((mN)^{O(1)})$ using the algorithm in [Aro96, Mit96]. An inspection of the analysis of greedy heuristic for weighted set cover (see, e.g., [Chv79, Hoc97]) reveals that this extra constant factor only increases the constant hidden in $O(\log m + \log N)$ $\square$

Algorithm *Greedy*
Input:

(i) The set of faces S in the subdivision.

(ii) The t-loci and t-regions for all tools.

(iii) The groups $G_i(t)$ of fixed points to be visited for all tools. Each fixed point is associated with a table keeping the cost and the number of faces covered by the visit. F_t denotes the maximum number of faces in some table entry in some $G_i(t)$.

Output: An approximate milling plan

while $S \neq \emptyset$ $\quad$ (* Loop until all faces are covered *)
    do for each tool t $\quad$ (* Find min avg weight tour for each tool *)
        do for $1 \leq j \leq F_t$
            do for each $G_i(t)$
                do solve the j-WTSP on $G_i(t)$;
            $T_j(t)$ = the tour with minimum weight among the j-WTSP solutions;
        $T(t)$ = minimum average weight $\{T_j(t)/j\}$;
    Set T to be the tour of minimum average weight among $T(t)$ for all tools t;
    output T as a tour in the output milling plan;
    remove faces covered in T from S; $\quad$ (* Update for next iteration *)
    update F_t for all t;
    update the numbers of faces in all tables;

Table 1: Greedy heuristic.

Acknowledgments

We thank Ajay Joneja of the Industrial Engineering Department at the Hong Kong University of Science and Technology for suggesting the multiple-tool milling problem and for valuable discussions on the practical aspects of milling.

References

[AFM93] E. Arkin, S. Fekete, and J. Mitchell. The lawnmower problem. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 461–466, 1993.

[AFM95] E. Arkin, S. Fekete, and J. Mitchell. Approximation algorithms for lawnmowing and milling. Report, SUNY Stony Brook, 1995.

[AHS96] E. M. Arkin, M. Held, and C. L. Smith. Optimization problems related to zigzag pocket machining. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 419–428, 1996.

[Aro96] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 2–11, 1996.

[Aro97] S. Arora. Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. 38th Annu. IEEE Sympos. Found. Comput. Sci.*, 1997.

[BC91] M. Bala and T.-C. Chang. Automatic cutter selection and optimal cutter path generation for prismatic parts. *Int. J. Prod. Res*, 29:2163–2176, 1991.

[BKSV97] M. de Berg, M. Katz, A. F. van der Stappen, and J. Vleugels. Realistic input models for geometric algorithms. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 294–303, 1997.

[Bru82] L. K. Bruckner. Geometric algorithms for 2.5d roughing process of sculptured surfaces. In *Proc. Joint Anglo-Hungarian Seminar on Computer-Aided Geometric Design*, 1982.

[Chv79] V. Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4:233–235, 1979.

[DBW93] A. V. Deshmukh, M. M. Barash, and H. P. Wang. On selection of tool path orientations for generating prismatic features. Report, IE Dept, Purdue U, 1993.

[HA92] A. Hansen and F. Arbab. An algorithm for generating NC tool paths for arbitrarily shaped pockets with islands. *ACM Trans. Graph.*, 11(2):152–182, April 1992.

[Hel89] M. Held. Geopocket – A sophisticated computational geometry solution of geometrical and technological problems arising from pocket machining. In F. Kimura and A. Rolstadås, editors, *Proc. Computer Applications in Production and Engineering*, pages 283–293. North-Holland, October 1989.

[Hel91a] M. Held. A geometry-based investigation of the tool path generation for zigzag pocket machining. *Visual Comput.*, 7(5–6):296–308, September 1991.

[Hel91b] M. Held. *On the Computational Geometry of Pocket Machining*, volume 500 of *Lecture Notes Comput. Sci.* Springer-Verlag, June 1991.

[Hoc97] D. S. Hochbaum. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, Boston, MA, 1997.

[IRT94] K. Iwano, P. Raghavan, and H. Tamaki. The traveling cameraman problem, with applications to automatic optical inspection. In *Proc. 5th Annu. Internat. Sympos. Algorithms Comput.*, Lecture Notes Comput. Sci., Beijing, 1994. Springer-Verlag.

[Kra92] T. R. Kramer. Pocket milling with tool engagement detection. *J. of Maufacturing Systems*, 11(2):114–123, 1992.

[Mit96] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric k-MST problem. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 402–408, 1996.

[MMS94] J. S. B. Mitchell, D. M. Mount, and S. Suri. Query-sensitive ray shooting. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 359–368, 1994.

[Per78] H. Persson. NC machining of arbitrarily shaped pockets. *Comput. Aided Design*, 10(3):169–174, May 1978.

[PGW90] P. V. Prabhu, A. K. Gramopadhye, and H. P. Wang. General mathematical model to optimizing NC tool path for face milling of flat convex polygonal surfaces. *Int. J. Production Research*, 28(1):101–130, 1990.

[PK83] K. Preiss and E. Kaplansky. Automatic mill routing from solid geometry information. In *Proc. Computer Applications in Production and Engineering.* North-Holland, 1983.

[PK85] K. Preiss and E. Kaplansky. Automated CNC milling by artificial intelligence methods. *J. of Maufacturing Systems*, 4(1):51–63, 1985.

[Pre89] K. Preiss. Automated mill pocketing computations. In *Advanced Geometric Modeling for Engineering Applications*. North-Holland, 1989.

[SL90] Y. S. Suh and K. Lee. NC milling tool path generation for arbitrary pockets defined by sculptured surfaces. *Comput. Aided Design*, 22(5):273–284, June 1990.

[W$^+$87] H.-P. Wang et al. On the efficiency of NC tool path planning for face milling operations. *Trans. of the ASME, J. of Engineering for Industry*, 109(4):370–376, November 1987.