# Optimal Covering Tours with Turn Costs

Esther M. Arkin* Michael A. Bender[†] Erik D. Demaine[‡] Sándor P. Fekete[§]
Joseph S. B. Mitchell* Saurabh Sethia[†]

**Abstract**
We give the first algorithmic study of a class of "covering tour" problems related to the geometric Traveling Salesman Problem: Find a polygonal tour for a *cutter* so that it sweeps out a specified region ("pocket"), in order to minimize a cost that depends not only on the length of the tour but also on the number of *turns*. These problems arise naturally in manufacturing applications of computational geometry to automatic tool path generation and automatic inspection systems, as well as arc routing ("postman") problems with turn penalties. We prove lower bounds (NP-completeness of minimum-turn milling) and give efficient approximation algorithms for several natural versions of the problem, including a polynomial-time approximation scheme based on a novel adaptation of the $m$-guillotine method.
**Key Words:** NC machining, manufacturing, traveling salesman problem (TSP), milling, lawn mowing, covering, approximation algorithms, polynomial-time approximation scheme (PTAS), $m$-guillotine subdivisions, NP-completeness, turn costs.

## 1 Introduction

An important algorithmic problem in manufacturing is to compute effective paths and tours for covering ("milling") a given region ("pocket") with a cutting tool: Find a path or tour along which to move a prescribed cutter in order that the sweep of the cutter (exactly) covers the region, removing all of the material from the pocket, while not "gouging" the material that lies outside of the pocket. This covering tour problem and its variants arise not only in NC (Numerical Control) machining applications but also in several other applications, including automatic inspection, spray painting/coating operations, robotic exploration, arc routing, and even mathematical origami. While we will often speak of the problem as "milling" with a "cutter," many of its important applications arise in various contexts outside of machining.

The majority of research on these geometric covering tour problems as well as on the underlying arc routing problems in networks has focused on cost functions based on the lengths of edges. However, in many actual routing problems, this cost is dominated by the cost of switching paths or direction at a junction. A drastic example is given by fiber-optical networks, where the time to follow an edge is negligible compared to the cost of changing to a different frequency at a router. In the context of NC machining, turns are an important component of the objective function, as the cutter may have to be slowed in anticipation of a turn. The number of turns (or the "link distance") also arises naturally as an objective function in robotic exploration (minimum-link watchman tours) and in various arc routing problems (snow plowing or street sweeping with turn penalties).

In this paper, we address the problem of minimizing the cost of *turns* in a covering tour. This important aspect of the problem has been left unexplored so far in the algorithmic community; the arc routing community has examined only heuristics or exact algorithms that do not have performance guarantees. We provide several new results:

(1) We prove that the covering tour problem with turn costs is NP-complete, even if the objective is purely to minimize the number of turns, the pocket is orthogonal (rectilinear), and the cutter must move axis-parallel. The hardness of the problem is not apparent, as our problem seemingly bears a close resemblance to the polynomially-solvable Chinese postman problem; see the discussion below.

(2) We provide a variety of constant-factor approximation algorithms that efficiently compute covering tours that are nearly optimal with respect to turn costs in various versions of the problem. While getting *some* $O(1)$-approximation is not difficult for most problems in this class, through a careful study of the structure of the problem, we have

*Dept. of Applied Mathematics and Statistics, SUNY Stony Brook, NY 11794-3600, {estie, jsbm} @ams.sunysb.edu.

[†]Dept. of Computer Science, SUNY Stony Brook, NY 11794-4400, {bender, saurabh}@cs.sunysb .edu.

[‡]Dept. of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, eddemaine@uwaterloo.ca.

[§]Fachbereich Mathematik, TU Berlin, 10623 Berlin, Germany, fekete@math.tu-berlin.de.

| Milling problem | Cycle Cover APX | Tour APX | Simultaneous Length APX | Maximum Coverage |
|---|---|---|---|---|
| Discrete | $2\delta + \rho$ | $2\delta + \rho + 2$ | $\delta$ | $\mu$ |
| Restricted-direction | $5d$ | $5d + 2$ | $2d$ | $2d$ |
| Orthogonal | 4.5<br>10 | 6.25<br>12 | 8<br>4 | 8<br>4 |
| Integral orthogonal | 2.5 | 3.75 | 4 | 4 |
| Orthogonal thin | 1 | 4/3 | 4 | 4 |
| Eulerian orthogonal thin | 1 | 6/5 | 4 | 4 |

Table 1: Approximation factors achieved by our (polynomial-time) algorithms. Columns marked "APX" give approximation factors: "Cycle Cover" is with respect to a minimum-turn cycle cover, "Tour APX" for a minimum-turn covering tour, and "Simultaneous Length APX" for a minimum-length covering tour. "Maximum Coverage" indicates the maximum number of times a point is visited. $\delta$ and $\mu$ denote the average and the maximum degree in the underlying graph. $\rho$ is the average number of directions in the graph; $d$ indicates the number of directions in the restricted-direction scenario. (See Section 2 for more detailed definitions.)

developed tools and techniques that enable significantly stronger approximation results. One of our main results is a 3.75-approximation for minimum-turn axis-parallel tours for a unit square cutter that covers an integral orthogonal polygon (with holes). Another main result gives a 4/3-approximation for minimum-turn tours in a "thin" pocket, as arises in the arc routing version of our problem.
Table 1 summarizes our various results.

(3) We give a polynomial-time approximation scheme (PTAS) for the covering tour problem in which the cost is given as a weighted combination of length and number of turns; e.g., the Euclidean length plus a constant $C$ times the number of turns. For a polygon with $h$ holes, the running time is $O(2^h \cdot N^{O(C)})$. The PTAS involves an extension of the $m$-guillotine method, which has previously been applied to obtain PTAS's in problems involving only *length*.

**1.1 Related Work.** In the CAD community, there is a vast literature on the subject of automatic tool path generation; we refer the reader to Held [21] for a survey and for applications of computational geometry to the problem. The algorithmic study of the problem has focused on the problem of minimizing the length of a milling tour: Arkin et al. [5, 6] show that the problem is NP-hard in general. Constant-factor approximation algorithms are given in [5, 6, 23], with the best current factor being a 2.5-approximation for min-length milling (11/5-approximation for orthogonal simple polygons). For the closely related *lawn mowing* problem (also known as the "traveling cameraman problem" [23]), in which the covering tour is not constrained to stay within $P$, the best current approximation factor is $(3 + \varepsilon)$ (utilizing PTAS results for TSP). Also closely related is the watchman route problem with limited visibility (or "$d$-sweeper problem"), as studied by Ntafos [31], who provides a 4/3-approximation, which is improved to a 6/5-approximation by [6]. The problem is also closely related to the Hamiltonicity problem in grid graphs; the recent results of [32] suggest that in *simple* polygons, minimum-length milling may in fact have a polynomial-time algorithm.

Covering tour problems are related to *watchman route* problems in polygons, which have had considerable study in terms of both exact algorithms (for the simple polygon case) and approximation algorithms (in general); see [29] for a recent survey. Most relevant to our problem is the prior work on *minimum-link* watchman tours: see [2, 3, 8] for hardness and approximation results, and [14, 25] for combinatorial bounds. However, in these problems the watchman is assumed to see arbitrarily far, making them distinct from our tour cover problems.

Other algorithmic results on milling include a recent study of *multiple tool* milling by Arya, Cheng, and Mount [9], who give an approximation algorithm for minimum-length tours that use different size cutters, and a recent paper of Arkin et al. [7], who examine the problem of minimizing the number of retractions for "zig-zag" machining without "re-milling," showing that the problem is NP-complete and giving an $O(1)$-approximation algorithm.

Geometric tour problems with turn costs have been studied by Aggarwal et al. [1], who prove NP-complete the *angular-metric TSP*, in which one is to compute a tour on a set of points in order to minimize the sum of the direction changes at each vertex. Fekete [17] and Fekete and Woeginger [18] have studied a variety of *angle-restricted tour* (ART) problems.

In the operations research literature, there has been extensive work on *arc routing* problems, which are variations on the classic Chinese postman problem, arising in snow removal, street cleaning, road gritting, trash collection, meter reading, mail delivery, etc.; see the surveys of [10, 15, 16]. Arc routing with turn costs has had considerable attention recently, as it enables a more accurate modeling of the true routing costs in many situations. Most recently, Clossey et al. [13] present six heuristic methods of attacking arc routing with turn penalties, without resorting to the usual transformation to a TSP problem; however, their results are purely based on experiments and provide no provable performance guarantees. The *directed* postman problem with turn penalties has been studied recently by Benavent and Soler [11], who prove the problem to be (strongly) NP-hard and provide heuristics (without performance guarantees) and computational results. See also Soler's thesis [19] and [30] for computational experience with worst-case exponential-time exact methods.

Our covering tour problem is related to the Chinese postman problem, which is readily solved exactly in polynomial time. However, the turn-weighted Chinese postman problem is readily seen to be NP-complete: Hamiltonian cycle in line graphs is NP-complete (contrary to what is reported in [20]; see page 246, West [33]), implying that TSP in line graphs is also NP-complete. The Chinese postman problem on graph $G$ with turn costs at nodes (and zero costs on edges) is equivalent to TSP on the corresponding line graph, $\mathcal{L}(G)$, where the cost of an edge in $\mathcal{L}(G)$ is given by the corresponding turn cost in $G$. Thus, the turn-weighted Chinese postman problem is also NP-complete.

## 2 Preliminaries

**2.1 Problem Definitions.** The general *geometric milling problem* is to find a closed curve (not necessarily simple) whose Minkowski sum with a given tool (cutter) is precisely a given region (pocket), $P$. Subject to this constraint, we may wish to optimize a variety of objective functions, such as the length of the tour, or the number of turns in the tour. We call these problems *minimum-length* and *minimum-turn* milling, respectively. While the latter problem is the main focus of this paper, we are also interested in bicriteria versions of the problem in which both length and number of turns must be small, or some linear combination of the two (see Section 5.6).

In addition to choices in the objective function, the problem version depends on the constraints on the tour. In the *orthogonal milling problem*, the region $P$ is an orthogonal polygonal domain (with holes) and the tool is an (axis-parallel) unit-square cutter constrained to axis-parallel motion, with links of the tour alternating between horizontal and vertical. All turns are 90°; a "U-turn" has cost of 2.

Instead of dealing directly with a geometric milling problem, we often find it helpful to consider a more combinatorial problem, and then adapt the solution back to the geometric problem. The *integral orthogonal milling problem* is a specialization of the orthogonal milling problem in which the region $P$ has integral vertices. In this case, an optimal tour can be assumed to have its vertex coordinates of the form $k+\frac{1}{2}$ for integral $k$. Then, milling in an integral orthogonal polygon (with holes) is equivalent to finding a tour of all the vertices ("pixels") of a grid graph; see Figure 1.

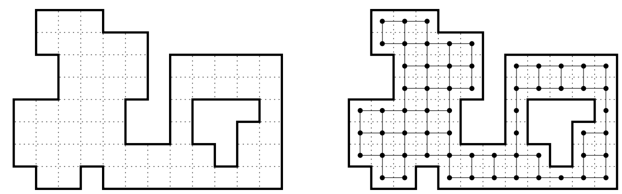

Figure 1: An instance of the integral orthogonal milling problem (left), and the grid graph model (right).

A more general combinatorial model than integral orthogonal milling is the *discrete milling problem*, in which we discretize the set of possible links into a finite collection of "channels" which are connected together at "vertices." More precisely, the channels have unit "width" so that there is only one way to traverse them with the given unit-size tool. At a vertex, the tour has a choice of (1) turning onto another channel connected at that end (costing one turn), (2) going straight if there is an incident channel collinear with the source channel (costing no turns), or (3) "U-turning" back onto the source edge (costing two turns). Hence, this problem can be modeled by a graph with certain pairs of incident edges marked as "collinear," in such a way that the set of collinear pairs at each vertex is a (not necessarily perfect) matching. The discrete milling problem is to find a tour in such a graph that visits every vertex. (The vertices represent the "pixels" to be covered.) Integral orthogonal milling is the special case of discrete milling in a grid graph. Let $\delta$ (resp., $\mu$) denote the average (resp., maximum) degree of a vertex and let $\rho$ denote the average number of distinct "directions" coming together at a vertex, that is, the average over each vertex of the cardinality of the matching plus the number of unmatched edges at that vertex.

The *thin milling problem* is to find a tour in such a graph that traverses every edge (and visits every vertex). Thus, the minimum-length version of thin milling is exactly the Chinese postman problem. As we have already noted, the minimum-turn version is NP-complete. The *orthogonal thin milling problem* is the special case in which the graph comes from an instance of orthogonal milling.

A generalization of orthogonal milling is the milling problem with a constant number $d$ of allowed directions, which we call *restricted-direction milling.* In particular, the region $P$ can only have edges with the $d$ allowed directions. This problem is not an immediate subproblem of discrete milling, since the cost function differs at vertices where "going straight" does not incur any cost; however, similar techniques can be used to achieve approximation algorithms.

**2.2 Other Issues.** It should be stressed that using turn cost instead of (or in addition to) edge length changes several characteristics of distances. One fundamental problem is illustrated by the example in Figure 2: the triangle inequality does not have to hold when using turn cost. This implies that many classical algorithmic approaches for graphs with nonnegative edge weights (such as using optimal 2-factors or the Christofides method for the TSP) cannot be applied without developing additional tools.

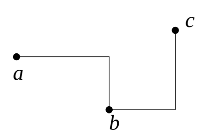


Figure 2: The triangle inequality may not hold when using turn cost as distance measure: $d(a, c) = 3 > 2 = d(a, b) + d(b, c)$

In fact, in the presence of turn costs we distinguish between the terms *2-factor* and *cycle cover*. While the terms are interchangeable when referring to the set of edges that they constitute, we make a distinction between their respective *costs*: a "2-factor" has a cost consisting of the sum of edge costs, while the cost of a "cycle cover" includes also the turn costs at vertices.

It is often useful in designing approximation algorithms for optimal *tours* to begin with the problem of computing an optimal *cycle cover*, minimizing the total number of turns in a set of cycles that covers $P$. Specifically, we can decompose the problem of finding an optimal (minimum-turn) tour into two tasks: finding an optimal cycle cover, and merging the components. Of course, these two processes may influence each other: there may be several optimal cycle covers, some of which are easier to merge than others. In particular, we say that a cycle cover is *connected* if the graph induced by the set of cycles and their intersections is connected. As we will show, even the problem of optimally merging a connected cycle cover is NP-complete. This is in contrast to minimum-length milling, where an optimal connected cycle cover can trivially be converted into an optimal tour that has the same cost.

Another important issue is the encoding of the input and output. In integral orthogonal milling, one might think that it is most natural to encode the grid graph, since the tour will be embedded on this graph and will, in general, have complexity proportional to the number of pixels. But the input to any geometric milling problem has a natural encoding by specifying the vertices of the polygon $P$. In particular, long edges are encoded in binary (or with one real number, depending on the model) instead of unary. It is possible to get a running time depending only on this size, but of course we need to allow for the output to be encoded *implicitly.* That is, we cannot explicitly encode each vertex of the tour, because there are too many (it can be arbitrarily large even for a succinctly encodable rectangle). Instead, we encode an abstract description of the tour that is easily decoded.

Algorithms whose running time is polynomial in the explicit encoding size (pixel count) are *pseudo-polynomial.* Algorithms whose running time is polynomial in the implicit encoding size are *polynomial.* For our purposes it will not matter whether lengths are encoded with a single real number or in binary.

Finally, we mention that many of our results carry over from the *tour* (or cycle) version to the *path* version, in which the cutter need not return to its original position. We omit discussion here of the changes necessary to compute optimal paths. We also omit in this abstract discussions of how our results apply also to the case of *lawn mowing*, in which the sweep of the cutter is allowed to go outside $P$ during its motion.

With so many problems of interest, we specify in every lemma, theorem, and corollary to which class of problems it applies. The default subproblem is to find a tour; if this is not the case (e.g., it is to find a cycle cover), we state it explicitly.

## 3 NP-Completeness

Arkin et al. [6] have proved that the problem of optimizing the length of a milling tour is NP-hard. This implies that it is NP-hard to find a tour of minimum total length that visits all vertices. If, on the other hand, we are given a connected cycle cover of a graph that has minimum total length, then it is trivial to convert it into a tour of the same length by merging the cycles into one tour.

We show that if the quality of a tour is measured by counting *turns*, then even this last step of turning an optimal connected cycle cover into an optimal tour is NP-complete. This implies NP-hardness of finding a milling tour that optimizes the number of turns for a polygon with holes.

Theorem 3.1. *Minimum-turn milling is NP-complete, even when we are restricted to the orthogonal thin case, and assume that we know an optimal (minimum-turn) connected cycle cover.*

See the full version of this paper [4] for proofs of this and most other theorems and lemmas. Because orthogonal thin milling is a special case of thin milling as well as orthogonal milling, and it is easy to convert an instance of thin orthogonal milling into an instance of integral orthogonal milling, we have

Corollary 3.1. *Discrete milling, restricted-direction milling, orthogonal milling, and integral orthogonal milling are NP-complete.*

## 4 Approximation Tools

There are three main tools that we use to develop approximation algorithms: computing optimal cycle covers for milling the "boundary" of $P$ (Section 4.1), converting cycle covers into tours (Section 4.2), and utilizing optimal (or nearly optimal) "strip covers" (Section 4.3).

**4.1 Boundary Cycle Covers.** We consider first the problem of finding a minimum-turn cycle cover for covering a certain subset, $\overline{P}$, of $P$ that is along its boundary. Specifically, in orthogonal milling we define the *boundary links* to be orthogonal offsets of each boundary edge, by 0.5 towards the interior of $P$, shrunk by 0.5 at each end. (In the nonorthogonal case, the notion of boundary link can be generalized; we defer the details to the full paper.) The region $\overline{P}$ is defined, then, to be the Minkowski sum of the boundary links and the tool, and we say that a cycle cover or tour *mills the boundary* if it covers $\overline{P}$. In the integral orthogonal case, $\overline{P}$ is the union of pixels having at least one edge against the boundary of $P$. We exploit the property that a cycle cover that mills the boundary (or a cycle cover that mills the entire region) can be assumed to include the boundary links in their *entirety* (without turns):

Lemma 4.1. *Any cycle cover that mills the boundary can be converted into one that includes each boundary link as a portion of a link, without changing the number of turns.*

This property allows us to apply methods similar to those used in solving the Chinese postman problem: we know portions of links that must be in the cycle cover, and furthermore these links mill the boundary. What remains is to connect these links into cycles, while minimizing the number of additional turns.

We can compute the "turn distance" (which is one less than the link distance) between an endpoint of one boundary link and an endpoint of another boundary link. The crucial knowledge that we are using is the orientation of the boundary links, so, for example, we correctly compute that the turn distance is zero when two boundary links are collinear and disjoint. Now we can find a minimum-weight perfect matching in the complete graph on boundary-link endpoints, with each edge weighted according to the corresponding turn distance. This connects the boundary links optimally into a set of cycles, proving

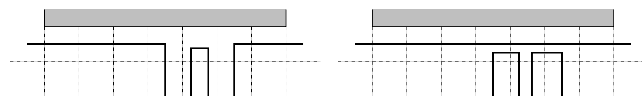


Figure 3: By performing local modifications, an optimal cycle cover can be assumed to cover each piece of the boundary in one connected link.

Theorem 4.1. *A min-turn cycle cover of the boundary of a region can be computed in polynomial time.*

**Remark.** Note that the definition of the "boundary" region $\overline{P}$ used here does not include all pixels that touch the boundary of $P$; in particular, it omits the "reflex pixels" that share a corner, but no edge, with the boundary of $P$. It seems difficult to require that the cycle cover mill reflex pixels, since Lemma 4.1 does not extend to this case, and an optimal cycle cover of the boundary (as defined above) may have fewer turns than an optimal cycle cover that mills the boundary $\overline{P}$ plus the reflex pixels; see Figure 4.

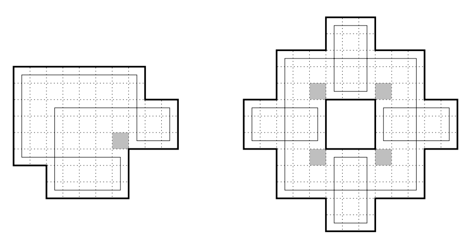

Figure 4: Optimally covering pixels that have an edge against the boundary can leave reflex pixels uncovered.

**4.2 Merging Cycles.** It is often easier to find a minimum-turn cycle cover (or constant-factor approximation thereof) than to find a minimum-turn tour. Here, we show that an exact or approximate minimum-turn cycle cover implies an approximation for minimum-turn tours.

Theorem 4.2. *A cycle cover with $t$ turns can be converted into a tour with at most $t + 2c - 2$ turns, where $c$ is the number of cycles.*

*Proof Idea.* By breaking two cycles into paths, it is possible to extend the end links to join pairs from opposite cycles, introducing two new turns. □

COROLLARY 4.1. *A cycle cover of a connected orthogonal polygon with $t$ turns can be converted into a single milling tour with at most $\frac{3}{2}t$ turns.*

*Proof.* Follows immediately from Theorem 4.2 and the fact that each cycle has at least four turns. □

Unfortunately, general merging is difficult (as illustrated by the NP-hardness proof), so we cannot hope to improve these general merging results by more than a constant factor.

**4.3 Strip and Star Covers.** A key tool for approximation algorithms is a covering of the region by a collection of "strips." In general, a *strip* is a maximal link whose Minkowski sum with the tool is contained in the region. A *strip cover* is a collection of strips whose Minkowski sums with the tool cover the entire region. A *minimum strip cover* is a strip cover with the fewest strips.

LEMMA 4.2. *The size of a minimum strip cover is a lower bound on the number of turns in a cycle cover (or tour) of the region.*

*Proof.* Any cycle cover induces a strip cover by extending each link to have maximal length. □

In the discrete milling problem, a related notion is a "rook placement." A *rook* is just a vertex that can *attack* each vertex to which it is connected via a single link. A *rook placement* is a collection of rooks no two of which can attack each other.

LEMMA 4.3. *The size of a maximum rook placement is a lower bound on the number of turns in a cycle cover (or tour) for discrete milling.*

*Proof Idea.* To get from one rook to another requires a turn. □

In the integral orthogonal milling problem, the notions of strip cover and rook placement are dual and efficient to compute:

LEMMA 4.4. *For integral orthogonal milling, a minimum strip cover and a maximum rook placement have equal size. Furthermore, they can be computed in time $O(n^{2.5})$.*

*Proof.* Consider the following graph $G$: Each vertex represents a maximal strip, i.e., a contiguous row or column or pixels. Two vertices are connected, iff the corresponding strips share a pixel. Clearly, this graph is bipartite, since rows only intersect columns, and columns only intersect rows. Now it is not hard to see that a minimum strip cover corresponds to a minimum vertex cover, while a maximum rook placement corresponds to a maximum matching. By the famous König-Egerváry theorem for matchings and vertex covers in bipartite graphs, these have the same value. Moreover, they can be computed in time $O(n^{2.5})$, as claimed. □

For general discrete milling, it is possible to approximate an optimal strip cover as follows. Greedily place rooks until no more can be placed (i.e., until there is no unattackable vertex). This means that every vertex is attackable by some rook, so by replacing each rook with all possible strips through that vertex, we obtain a strip cover of size $\rho$ times the number of rooks. (We call this type of strip cover a *star cover.*) But each strip in a minimum strip cover can only cover a single rook, so this is a $\rho$-approximation to the minimum strip cover. We have thus proved

LEMMA 4.5. *In discrete milling, the number of stars in a greedy star cover is a lower bound on the number of strips, and hence serves as an $\rho$-approximation algorithm for minimum strip covers.*

It is also easy to show

LEMMA 4.6. *A greedy star cover can be found in linear time.*

## 5 Approximation Algorithms

**5.1 Discrete Milling.** Our most general approximation algorithm for the discrete milling problem has the additional feature of running in linear time. First we take a star cover according to Lemma 4.5 which approximates an optimal strip cover within a factor of $\rho$. Then we tour the stars carefully, depending on whether a channel continues straight through the star center or terminates there.

THEOREM 5.1. *For min-turn cycle covers in discrete milling, there is a linear-time $(\rho + 2\delta)$-approximation. Furthermore, the maximum coverage of a vertex (i.e., the maximum number of times a vertex is swept) is $\mu$, and the cycle cover is a $\delta$-approximation on length.*

By merging these tours using Theorem 4.2, we get the following:

COROLLARY 5.1. *For min-turn discrete milling, there is a linear-time $(\rho + 2\delta + 2)$-approximation. Furthermore, the maximum coverage of a vertex is $\mu$, and the tour is an $\delta$-approximation on length.*

Note that, in particular, these algorithms give a 6-approximation on length if the discrete milling problem comes from a planar graph, since the average degree $\delta$ of a planar graph is bounded by 6.

COROLLARY 5.2. *For minimum-turn integral orthogonal milling, there is a linear-time 12-approximation that covers each pixel at most 4 times and hence is also a 4-approximation on length.*

*Proof.* In this case, $\rho = 2$ and $\delta = \mu = 4$. □

**5.2 Restricted-Direction Milling.** We already mentioned above that restricted-direction milling is not an immediate special case of discrete milling; however, the same method and result applies:

THEOREM 5.2. *In restricted-direction milling, there is a $5d$-approximation on minimum-turn cycle covers that is linear in the number $N$ of pixels, and a $(5d + 2)$-approximation on minimum-turn tours of same complexity. In both cases, the maximum coverage of a point is at most $2d$, so the algorithms are also $2d$-approximations on length.*

Note that this approximation algorithm applies to geometric milling problems in arbitrary dimensions provided that the number of directions is bounded, e.g., orthogonal milling in 3-D.

As mentioned in the preliminaries, just the pixel count $N$ may not be a satisfactory measure for the complexity of an algorithm, as the original region may be encoded more efficiently by its boundary, and a tour may be encoded by structuring it into a small number of pieces that have a short description. It is possible to use the above ideas for approximation algorithms in this extended framework. For simplicity, we describe how this can be done for the integral orthogonal case.

THEOREM 5.3. *There is factor 10 approximation algorithm of (strongly polynomial) complexity $O(n \log n)$ on minimum-turn cycle cover for a region of pixels bounded by $n$ integral axis-parallel segments, and a 12-approximation on minimum-turn tours of same complexity. In both cases, the maximum coverage of a point is at most 4, so the algorithms are also 4-approximations on length.*

*For the special case in which the boundary is connected (the region has no holes), the complexities drop to $O(n)$.*

**5.3 Integral Orthogonal.** We have already shown a 12-approximation for minimum-turn integral orthogonal milling, using a star cover, with a running time of $O(N)$ or $O(n \log n)$. If we are willing to invest more time for computation, we can find an optimal rook cover (instead of a greedy one), which by Lemma 4.4 yields an optimal strip cover. This leads to

THEOREM 5.4. *There is an $O(n^{2.5})$-time algorithm that computes a milling tour with number of turns within 6 times the optimal, and with length within 4 times the optimal.*

*Proof Idea.* Form a cycle cover by "doubling" each strip, using two U-turns each, then merge. □

By more sophisticated merging procedures, it may be possible to reduce this approximation factor to something between 4 and 6. However, our best approximation algorithm uses a different strategy.

THEOREM 5.5. *There is an $O(n^{2.5})$ 2.5-approximation algorithm for minimum-turn cycle covers, and hence a polynomial-time 3.75-approximation for minimum-turn tours, for integral orthogonal milling.*

*Proof.* As described in Lemma 4.4, find an optimal strip cover $S$. Let $s$ be its cardinality, so $\text{OPT} \geq s$.

Now consider the end vertices of the strip cover. By construction, they are part of the boundary. For each end point of a strip, a tour either crosses it orthogonally, or the tour turns at the boundary segment. In any case, a tour must have a link that crosses an end vertex orthogonally to the strip. (Note that this link has zero length in case of a U-turn.)

Next consider the following distance function between end points of strips: For any pair of end points $u$ and $v$ (possibly of different strips $s_u$ and $s_v$), let $w(u, v)$ be the smallest number of links from $u$ to $v$ when leaving $u$ in a direction orthogonal to $s_u$, and arriving at $v$ in a direction orthogonal to $s_v$. By a standard argument, an optimal matching $M$ satisfies $w(M) \leq \text{OPT}/2$.

By construction, the edges of $M$ and the strips of $S$ induce a 2-factor of the end points. Since any matching edges leaves a strip orthogonally, we get at most 2 additional turns at each strip for turning each 2-factor into a cycle. The total number of turns is $2s + w(M) \leq 2.5\text{OPT}$. Since the strips cover the whole region, we get a feasible cycle cover.

Finally, we can use Corollary 4.1 to turn the cycle cover into a tour. By the corollary, this tour does not have more than $3.75\text{OPT}$ turns. □

The simple class of examples in Figure 5 shows that the cycle cover algorithm may use $2\text{OPT}$ turns, and the

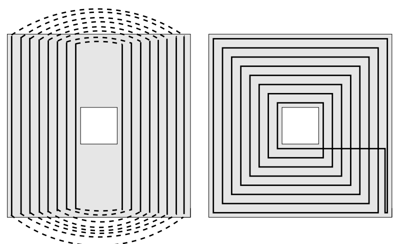


Figure 5: A bad example for the 3.75 approximation algorithm: (a) half the cycles constructed by the algorithm; (b) an optimal tour.

tour algorithm may use 3OPT turns, assuming that no special algorithms are used for matching and merging.

It consists of a "square donut" of width $k$. An optimal strip cover consists of $4k$ strips; an optimal matching of strip ends yields a total of $8k + 2$ turns, and we get a total of $2k$ cycles. (In Figure 5(a), only the vertical strips and their matching edges are shown to keep the drawing cleaner.) If the merging of these cycles is done badly (by merging cycles at crossings and not at parallel edges), it may cost another $4k - 2$ turns, for a total of $12k$ turns. As can be seen from Figure 5(b), there is a feasible tour that uses only $4k + 2$ turns. This shows that optimal tours may have almost all turns strictly inside of the region. Moreover, the same example shows that this 3.75-approximation algorithm does not give an immediate length bound on the resulting tour. However, we can use a local modification argument to show the following:

THEOREM 5.6. *For any given tour with $L$ links that mills an integral orthogonal region, we can perform at most $L$ local modifications, such that we get a tour with an equal number of turns that covers each pixel at most four times. This implies a performance ratio of 4 on the total length.*

### 5.4 Nonintegral Orthogonal Polygons.

Nonintegral orthogonal polygons present a difficulty in that no polynomial-time algorithm is known to compute a minimum strip cover for such polygons. Fortunately, however, we can use the boundary tours from Section 4.1 to get a better approximation algorithm than the 12 from Theorem 5.2.

THEOREM 5.7. *In nonintegral orthogonal milling, there is a polynomial-time 4.5-approximation for minimum-turn cycle covers and 6.25-approximation for minimum-turn tours. The running time is $O(n^{2.5})$.*

*Proof Idea.* Combine the 3.75-approximation (for covering the interior) with a boundary tour modified to cover subpixels near reflex vertices. □

### 5.5 Milling Thin Pockets.

In this section we consider the special case of milling *thin* integral orthogonal pockets, which contain no points on the integer lattice. Intuitively, a thin pocket is composed of a network of width-1 corridors.

We already know from Theorem 4.1 that a minimum-turn cycle cover of a thin region can be found in polynomial time, because any cycle cover of the boundary can be turned into a cycle cover of the entire thin region. By also applying Theorem 4.2, we immediately obtain

COROLLARY 5.3. *In thin milling, there is a polynomial-time algorithm for computing a min-turn cycle cover, and a polynomial-time 1.5-approximation for min-turn tours.*

More interesting is that we can do much better than general merging in the case of thin milling. The idea is to decompose the induced graph into a number of cheap cycles, and a number of paths.

**Milling Thin Eulerian Pockets.** We first solve the special case of milling Eulerian pockets, that is, pockets that can be milled without retractions, so that each edge is traversed by the cutting tool exactly once.

Although one might expect that the minimum-turn milling is one of the possible Eulerian tours of the graph, this is not always true; see the full version [4] for a class of examples with this property.

However, we can achieve the following approximation:

THEOREM 5.8. *There is a linear-time 6/5-approximation algorithm for minimum-turn milling in a thin Eulerian pocket.*

**Milling Arbitrary Thin Pockets.** Now we consider the case of general thin pockets. It turns out that the following merging algorithm does significantly better than general merging:

1. Find an optimal cycle cover.
2. Repeat until there is only one cycle in the cycle cover:
   (a) If there are any two cycles that can be merged without any extra cost, perform the merge.
   (b) Find a vertex at which two cycles cross each other.
   (c) Modify the vertex to incorporate at most two additional turns, thereby connecting the two cycles.

THEOREM 5.9. *The above algorithm finds a tour of length at most* 4/3·OPT.

The example in Figure 6 shows that the estimate for the performance ratio is tight.

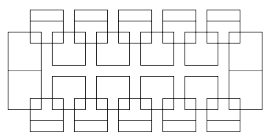

Figure 6: An example with performance ratio 4/3.

It consists of $k = 2s + 4$ cycles, all with precisely 4 turns, $s$ subtours without degree three vertices, and $s+4$ subtours with two degree three vertices each. We get opt $= 12s + 32$ and heur $= 16s + 38$, for a performance ratio arbitrarily close to 4/3 for large $s$.

**5.6 PTAS.** Here we outline a PTAS for the problem of minimizing a weighted average of the two cost criteria: length and number of turns. Our technique is based on using the theory of $m$-guillotine subdivisions [28], properly extended to handle turn costs. We prove the following result:

THEOREM 5.10. *For any fixed* $\varepsilon > 0$, *there is a* $(1 + \varepsilon)$*-approximation algorithm with running time* $O(2^h \cdot N^{O(C)})$ *that computes a milling tour for an integral orthogonal polygon* $P$ *with* $h$ *holes, where the cost of the tour is its length plus* $C$ *times the number of (90-degree) turns, and* $N$ *is the number of pixels in* $P$.

*Proof.* (sketch) Let $T^*$ be a minimum-cost tour. Following the notation of [28], we first use the main structure theorem to show that there exists an $m$-guillotine subdivision, $T_G$, obtained from $T^*$, with length at most $(1 + \frac{1}{m})$ times the length of $T^*$. (Note that part of $T_G$ may lie outside the pocket $P$, since we added $m$-spans to make it $m$-guillotine.) We then convert $T_G$ into a new graph, $T'_G$, which has the added properties that: (1) $T'_G \subset P$ (since we keep only those portions of the $m$-span that lie inside $P$), (2) the number of edges of $T^*$ incident on each component of the $m$-span is *even* (since $T^*$ is a tour), and (3) the total *cost* of $T'_G$ is at most $(1 + \frac{1+C}{m})$ times the cost of $T^*$. We then use dynamic programming to obtain a min-cost (modified) $m$-guillotine subdivision, $T^*_G$, which has certain specified properties, including connectedness, coverage, "bridge-doubling," and an even number of edges incident on each connected component of each $m$-span (this is the source of the $2^h$ term in the running time, as we need to be able to require a given parity at each sub-bridge, in order to have a connected Eulerian graph in the end, from which we can extract a tour). The techniques of [27] can be applied to reduce the exponent on $N$ to a term independent of $\varepsilon$. □

We expect to be able to use the same methods to obtain a PTAS that is polynomial in $n$ (versus $N$), with a careful consideration of implicit encodings of tours. We have not yet been able to give a PTAS for minimizing only the number of turns in a covering tour; this remains an intriguing open problem.

## 6 Conclusion

Many open problems remain:

1. Can we find a minimum-turn cycle cover in polynomial time? This would immediately lead to a 1.5-approximation for the orthogonal case, and a $(1+\frac{2}{3})$-approximation for the general case.
   Note that for the angular-metric TSP, finding a minimum-cost cycle cover for a planar set of points is NP-complete [1].
2. Is there a polynomial-time algorithm for exactly computing a minimum-turn covering tour for *simple* orthogonal polygons?
   The related problem for cost corresponding to distances is still open, but there is some evidence that it is indeed polynomial.
3. Is the analysis of the 3.75-approximation algorithm tight?
   There may be some redundancy in the combination of all the estimates; however, our example shows that even one of the basic simplifications (considering only maximal strips with endpoints on the boundary) may lead to a factor 2 or 3, depending on how the merging is done.
4. What is the complexity of computing a minimum strip cover in nonintegral orthogonal polygons?
5. What is the complexity of computing minimum strip covers in nonorthogonal polygons?
6. Is there a strip cover approximation algorithm for $d$ directions whose performance is independent of $d$?
7. Can one obtain approximation algorithms for unrestricted directions in an arbitrary polygonal domain?


## Acknowledgments

We thank Regina Estkowski for helpful discussions. This research is partially supported by NSF grant CCR-9732221, and by grants from Bridgeport Machines, HRL Laboratories, ISX Corporation, NASA Ames, Northrop-Grumman, Sandia National Labs, Seagull Technology, and Sun Microsystems.

## References


[1] A. Aggarwal, D. Coppersmith, S. Khanna, R. Motwani, and B. Schieber. The angular-metric traveling salesman problem. In *Proc. 8th Ann. ACM-SIAM Sympos. on Discrete Algorithms*, pp. 221–229, 1997.

[2] M. H. Alsuwaiyel and D. T. Lee. Minimal link visibility paths inside a simple polygon. *Comput. Geom. Theory Appl.*, **3(1)**, pp. 1–25, 1993.

[3] M. H. Alsuwaiyel and D. T. Lee. Finding an approximate minimum-link visibility path inside a simple polygon. *Inform. Process. Lett.*, **55(2)**, pp. 75–79, 1995.

[4] E. M. Arkin, M. A. Bender, E. D. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia. Optimal covering tours with turn costs. Technical Report 674, TU Berlin, 2000. Available at `http://www.math.tu-berlin.de/coga/publications/techreports/2000/Report-674-2000.html`.

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

[6] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, to appear.

[7] E. M. Arkin, M. Held, and C. L. Smith. Optimization problems related to zigzag pocket machining. *Algorithmica*, **26(2)**, pp. 197–236, 2000.

[8] E. M. Arkin, J. S. B. Mitchell, and C. Piatko. Minimum-link watchman tours. Technical report, SUNY Stony Brook, 1994.

[9] S. Arya, S.-W. Cheng, and D. Mount. Approximation algorithms for multiple-tool milling. In *Proc. 14th Ann. ACM Sympos. Comput. Geom.*, pp. 297–306, 1998.

[10] A. A. Assad and B. L. Golden. Arc routing methods and applications. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pp. 375–483. Elsevier Science, Amsterdam, 1995.

[11] E. Benavent and D. Soler. The directed rural postman problem with turn penalties. *Transporation Science*, **33(4)**, pp. 408–418, Nov. 1999.

[12] B. Chazelle. Triangulating a simple polygon in linear time. *Disc. Comput. Geom.*, **6(5)**, pp. 485–524, 1991.

[13] J. Clossey, G. Laporte, and P. Soriano. Solving arc routing problems with turn penalties. Technical Report G-2000-05, Le Groupe d'études et de recherche en analyse des décisions, Montréal, Canada, 2000.

[14] M. J. Collins and B. M. E. Moret. Improved lower bounds for the link length of rectilinear spanning paths in grids. *Inform. Process. Lett.*, **68**, pp. 317–319, 1998.

[15] H. A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems, part I: The Chinese postman problem. *Operations Research*, **43(2)**, pp. 231–242, 1995.

[16] H. A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems, part II: The rural postman problem. *Operations Research*, **43(3)**, pp. 399–414, 1995.

[17] S. P. Fekete. *Geometry and the Travelling Salesman Problem.* Ph.D. thesis, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1992.

[18] S. P. Fekete and G. J. Woeginger. Angle-restricted tours in the plane. *Comput. Geom. Theory Appl.*, **8(4)**, pp. 195–218, 1997.

[19] D. S. Fernández. *Problemas de Rutas por Arcos con Giros Prohibidos.* Ph.D. thesis, Universitat de Valencia, 1995.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979.

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

[22] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter. Hamilton paths in grid graphs. *SIAM J. Comput.*, **11**, pp. 676–686, 1982.

[23] 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.*, Vol. 834 *LNCS*, pp. 29–37. Springer-Verlag, 1994.

[24] D. S. Johnson and C. H. Papadimitriou. Computational complexity and the traveling salesman problem. In E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, editors, *The Traveling Salesman Problem*, pp. 68–74. John Wiley & Sons, New York, 1985.

[25] E. Kranakis, D. Krizanc, and L. Meertens. Link length of rectilinear Hamiltonian tours on grids. *Ars Combinatoria*, **38**, pp. 177–192, 1994.

[26] E. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart & Winston, New York, 1976.

[27] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: Part III – Faster polynomial-time approximation schemes for geometric network optimization. Manuscript, SUNY Stony Brook, 1997.

[28] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, $k$-MST, and related problems. *SIAM J. Comput.*, **28(4)**, pp. 1298–1309, 1999.

[29] J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 15, pp. 633–701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[30] E. M. Molada and D. S. Fernández. Exact solution of the Chinese postman problem with turn penalties. In *XV Congress on Differential Equations and Applications/V Congress on Applied Mathematics*, volume I, II (Spanish), Colecc. Congr., 9, pp. 1111–1115, 1998.

[31] S. Ntafos. Watchman routes under limited visibility. *Comput. Geom. Theory Appl.*, **1(3)**, pp. 149–170, 1992.

[32] C. Umans and W. Lenhart. Hamiltonian cycles in solid grid graphs. In *Proc. 38th Annu. IEEE Sympos. Found. Comput. Sci.*, pp. 496–507, 1997.

[33] D. West. *An Introduction to Graph Theory.* Prentice Hall, 1995.