The Lawn Mowing Problem: From Algebra to Algorithms

Sándor P. Fekete ✉ 🄳
Department of Computer Science, TU Braunschweig, Germany

Dominik Krupke ✉ 🄳
Department of Computer Science, TU Braunschweig, Germany

Michael Perk ✉ 🄳
Department of Computer Science, TU Braunschweig, Germany

Christian Rieck ✉ 🄳
Department of Computer Science, TU Braunschweig, Germany

Christian Scheffer ✉ 🄳
Faculty of Electrical Engineering and Computer Science, Bochum University of Applied Sciences, Bochum, Germany

──── **Abstract** ────

For a given polygonal region P, the Lawn Mowing Problem (LMP) asks for a shortest tour T that gets within Euclidean distance $1/2$ of every point in P; this is equivalent to computing a shortest tour for a unit-diameter cutter C that covers all of P. As a generalization of the Traveling Salesman Problem, the LMP is NP-hard; unlike the discrete TSP, however, the LMP has defied efforts to achieve exact solutions, due to its combination of combinatorial complexity with continuous geometry.

We provide a number of new contributions that provide insights into the involved difficulties, as well as positive results that enable both theoretical and practical progress. (1) We show that the LMP is algebraically hard: it is not solvable by radicals over the field of rationals, even for the simple case in which P is a 2×2 square. This implies that it is impossible to compute exact optimal solutions under models of computation that rely on elementary arithmetic operations and the extraction of kth roots, and explains the perceived practical difficulty. (2) We exploit this algebraic analysis for the natural class of polygons with axis-parallel edges and integer vertices (i.e., polyominoes), highlighting the relevance of turn-cost minimization for Lawn Mowing tours, and leading to a general construction method for feasible tours. (3) We show that this construction method achieves theoretical worst-case guarantees that improve previous approximation factors for polyominoes. (4) We demonstrate the practical usefulness *beyond polyominoes* by performing an extensive practical study on a spectrum of more general benchmark polygons: We obtain solutions that are better than the previous best values by Fekete et al., for instance sizes up to 20 times larger.

1 Introduction

Many geometric optimization problems are NP-hard: the number of possible solutions is finite, but there may not be an efficient method for systematically finding a best one. A different kind of difficulty considered in geometry is rooted in the impossibility of obtaining solutions with a given set of construction tools: Computing the length of a diagonal of a square is not possible with only rational numbers; trisecting any given angle cannot be done with ruler and compass, and neither can a square be constructed whose area is equal to that of a given circle.

In this paper, we consider the *Lawn Mowing Problem* (LMP), in which we are given a polygonal region P and a disk cutter C of diameter 1; the task is to find a closed roundtrip of minimum Euclidean length, such that the cutter "mows" all of P, i.e., a shortest tour that moves the center of C within distance $1/2$ from every point in P. The LMP naturally occurs in a wide spectrum of practical applications, such as robotics, manufacturing, farming, quality control, and image processing, so it is of both theoretical and practical importance. As a generalization of the classic Traveling Salesman Problem (TSP), the LMP is also NP-hard; however, while the TSP has shown to be amenable to exact methods for computing provably optimal solutions even for large instances [1], the LMP has defied such attempts, with only recently some first practical progress by Fekete et al. [26].

1.1 Related Work

There is a wide range of practical applications for the LMP, including manufacturing [5, 31, 32], cleaning [12], robotic coverage [13, 15, 29, 35], inspection [21], CAD [20], farming [6, 16, 40], and pest control [9]. In Computational Geometry, the Lawn Mowing Problem was first introduced by Arkin et al. [3], who later gave the currently best approximation algorithm with a performance guarantee of $2\sqrt{3}\alpha_{\text{TSP}} \approx 3.46\alpha_{\text{TSP}}$ [4], where α_{TSP} is the performance guarantee for an approximation algorithm for the TSP.

Optimally covering even relatively simple regions such as a disk by a set of n *stationary* unit disks has received considerable attention, but is excruciatingly difficult; see [10, 11, 28, 33, 36, 38]. As recently as 2005, Fejes Tóth [22] established optimal values for the maximum radius of a disk that can be covered by $n = 8, 9, 10$ unit circles. Recent progress on covering by (not necessarily equal) disks has been achieved by Fekete et al. [23, 24].

A first *practical* breakthrough on computing provably good Lawn Mowing tours was achieved by Fekete et al. [26], who established a primal-dual algorithm for the LMP by iteratively covering an expanding *witness set* of finitely many points in P. In each iteration, their method computes a lower bound, which involves solving a special case of a TSP instance with neighborhoods, the *Close-Enough TSP* (CETSP) to provable optimality; for an upper bound, the method is enhanced to provide full coverage. In each iteration, this establishes both a valid solution and a valid lower bound, and thereby a bound on the remaining optimality gap. They also provided a computational study, with good solutions for a large spectrum of benchmark instances with up to 2000 vertices. However, this approach encounters scalability issues for larger instances, due to the considerable number of witnesses that need to be placed.

A seminal result on algebraic aspects of geometric optimization problems was achieved by Bajaj [7], who established algebraic hardness for the Fermat-Weber problem of finding a point in $\mathbb{R}^2$ that minimizes the sum of Euclidean distances to all points in a given set. Others have studied the Galois complexity for geometric problems like Graph Drawing or the Weighted Shortest Path Problem [8, 14, 39].

As we will see in the course of our algorithmic analysis the number of turns in a tour is of crucial importance for the overall cost; this has been previously studied by Arkin et al. [2] in a discrete setting. This objective is also of practical importance in the context of physical coverage, e.g., in the context of efficient drone trajectories [9].

1.2 Our Results

We provide a spectrum of new theoretical and practical results for the Lawn Mowing Problem.

- We prove that computing an optimal Lawn Mowing tour is algebraically hard, even for the case of mowing a 2×2 square by a unit-diameter disk, as it requires computing zeroes of high-order irreducible polynomials.
- We exploit the algebraic analysis to achieve provably good trajectories for polyominoes, based on the consideration of turn cost, and provide a method for general polygons.
- We show that this construction method achieves theoretical worst-case guarantees that improve previous approximation factors for polyominoes.
- We demonstrate the practical usefulness *beyond polyominoes* on a spectrum of more general benchmark polygons, obtaining better solutions than the previous values by Fekete et al. [26], for instance sizes up to 20 times larger.

1.3 Definitions

A (simple) *polygon P* is a (non-self-intersecting) shape in the plane, bounded by a finite number n of line segments. The *boundary* of a polygon P is denoted by ∂P. A *polyomino* is a polygon with axis-parallel edges and vertices with integer coordinates; any polyomino can be canonically partitioned into a finite number N of unit-squares, called *pixels*. A *tour* is a closed continuous curve $T : [0, 1] \to \mathbb{R}^2$ with $T(0) = T(1)$. The *cutter C* is a disk of diameter d, centered in its midpoint. Without loss of generality, we assume $d = 1$ for the rest of the paper. The *coverage* of a tour T with the disk cutter C is the Minkowski sum $T \oplus C$. A *Lawn Mowing tour T* of a polygon P with a cutter C is a tour whose coverage contains P. An *optimal* Lawn Mowing tour is a Lawn Mowing tour of shortest length.

2 Algebraic Hardness

In their recent work, Fekete et al. [26] prove that an optimal Lawn Mowing tour for a polygonal region is necessarily polygonal itself; on the other hand, they show that optimal tours may need to contain vertices with irrational coordinates corresponding to arbitrary square roots, even if P is just a triangle. In the following, we show that if P is a 2×2 square, an optimal tour may involve coordinates that cannot even be described with radicals. See Figure 2 for the structure of optimal trajectories.

▶ **Theorem 1.** *For the case in which P is a 2×2 square, the Lawn Mowing Problem is algebraically hard: an optimal tour involves coordinates that are zeroes of polynomials that cannot be expressed by radicals.*

2.1 Optimal Tours at Corners

For the 2×2 square P, consider the upper left 1×1 subsquare S_0 with corners $(0, 0)$, $(0, 1)$, $(1, 1)$, $(0, 1)$, further subdivided into four $1/2 \times 1/2$ quadrants $S_{0,0}, \ldots, S_{0,3}$, and an optimal path ω that enters S_0 at the bottom and leaves it to the right. Let $p_s = (p_s^x, 0)$, $p_t = (1, p_t^y)$
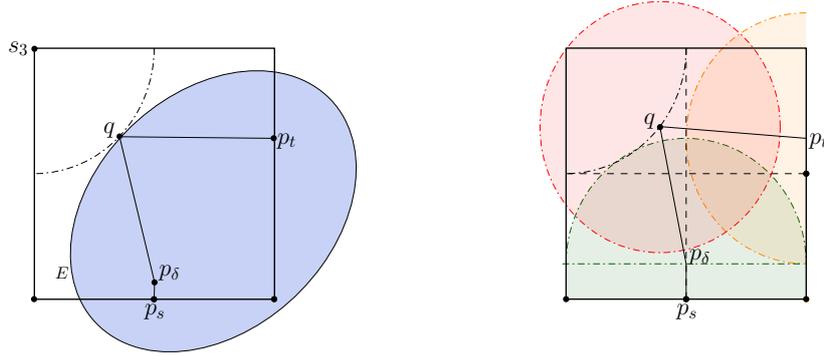
(a) Any $0 \leq \delta \leq 1$ defines p_δ, p_t, q and ellipse E.   **(b)** The optimal path ω through S_0.

Figure 1 Visualizations for Lemma 4.

be the points where ω enters and leaves S_0, respectively. For the following lemmas, we assume that a covering path exists that obeys the above conditions. We will later determine that path and show that it covers S_0. The proofs of Lemmas 2, 3, and 5 can be found in the full version [27].

▶ **Lemma 2.** $p_s^x \leq 1/2$ and $p_t^y \geq 1/2$ and either $p_s^x = 1/2$ or $p_t^y = 1/2$.

Without loss of generality, we assume that $p_s^x = 1/2$. The next step is to find the optimal position of p_t. As an optimal path ω must enter the quadrant $S_{0,3}$ once, we can subdivide the path into two parts. For some $\delta > 0$, let $p_t^y = 1/2 + \delta$ and $p_\delta = (1/2, \delta)$.

▶ **Lemma 3.** For any $\delta > 0$, ω has a subpath $p_s p_\delta$.

▶ **Lemma 4.** The uniquely-shaped optimal Lawn Mowing path ω between two adjacent sides of S_0 has length $L_{S_0} \approx 1.309$ with $\omega = (p_s, p_\delta, q, p_t)$ and

$$p_s = (1/2, 0) \quad p_\delta = (1/2, \delta) \approx (1/2, 0.168) \quad q \approx (0.386, 0.682) \quad p_t = (1, 1/2 + \delta) \approx (1, 0.668).$$

Proof. Let s_3 be the top left corner of S_0. We identify a shortest path for visiting one point q on a circle U with diameter 1 centered in s_3 dependent on δ, a necessary condition for a feasible path. Let $c = d(p_\delta, q) + d(q, p_t)$ be the distance from both points to U. Consider an ellipse E with foci p_δ, p_t that touches U in a single point, see Figure 1a. By definition, the intersection point q minimizes the distance c. For $\delta \in [0, 1]$ we want to find an intersection point between E and U that minimizes distance c. We can solve this problem with an exact optimization approach using Mathematica, see the full version of our paper [27]. It turns out that δ, q^x, q^y can only be expressed as the first, third, and first roots of three irreducible high-degree polynomials $f_\delta, f_{q^x}, f_{q^y}$, see Equation (1) and the full version [27].

$$\begin{aligned} f_\delta(x) = {} & 589\,824x^{16} - 7\,077\,888x^{15} + 41\,189\,376x^{14} - 154\,386\,432x^{13} + \quad (1) \\ & 416\,788\,480x^{12} - 857\,112\,576x^{11} + 1\,383\,417\,856x^{10} - 1\,779\,354\,624x^9 + \\ & 1\,834\,437\,632x^8 - 1\,514\,108\,928x^7 + 992\,782\,336x^6 - 509\,312\,064x^5 + \\ & 199\,354\,208x^4 - 57\,160\,752x^3 + 11\,200\,088x^2 - 1\,313\,928x + 67\,417 \end{aligned}$$

The value for $\delta \approx 0.167876$ defines the points p_δ and p_t. Together with the values for q^x, q^y, we obtain the path above. The combined length of the path is $\delta + c \approx 1.308838224$. As ω contains a subpath that crosses the full height of $S_{0,0}$ and another subpath that crosses the full width

of $S_{0,2}$, both quadrants are covered by ω, see Figure 1b. By construction, the bottom right quadrant is covered by the segment $p_s p_\delta$ and the point p_t. The top left quadrant is covered by q, because $S_{0,3}$ is fully contained in a disk with diameter 1 centered in q. Therefore, ω is a feasible path between two adjacent edges of S_0 with a length of $L \approx 1.309$. ◄

▶ **Lemma 5.** *A square P of side length 2 has a uniquely-shaped optimal Lawn Mowing tour T of length $L = 4L_{S_0}$, where $L_{S_0} \approx 1.309$.*

2.2 Galois Group of the Polynomial

Now we show that the coordinates of the optimal path ω can not be expressed by radicals. We employ a similar technique as Bajaj [7] for the generalized Weber problem. A *field K* is said to be an *extension* (written as $K/\mathbb{Q}$) of $\mathbb{Q}$ if K contains $\mathbb{Q}$. Given a polynomial $f(x) \in \mathbb{Q}[x]$, a finite extension K of $\mathbb{Q}$ is a *splitting field* over $\mathbb{Q}$ for $f(x)$ if it can be factorized into linear polynomials $f(x) = (x - a_1) \cdots (x - a_k) \in K[x]$ but not over any proper subfield of K. Alternatively, K is a splitting field of $f(x)$ of degree n over $\mathbb{Q}$ if K is a minimal extension of $\mathbb{Q}$ in which $f(x)$ has n roots. Then the *Galois group* of the polynomial f is defined as the Galois group of $K/\mathbb{Q}$. In principle, the Galois group is a certain permutation group of the roots of the polynomial. From the fundamental theorem of Galois theory, one can derive a condition for solvability by radicals of the roots of $f(x)$ in terms of algebraic properties of its Galois group. We state three additional theorems from Galois theory and Bajaj's work. The proofs can be found in [7, 34].

▶ **Lemma 6** ([34]). *$f(x) \in \mathbb{Q}[x]$ is solvable by radicals over $\mathbb{Q}$ iff the Galois group over $\mathbb{Q}$ of $f(x)$, $Gal(f(x))$, is a solvable group.*

▶ **Lemma 7** ([34]). *The symmetric group S_n is not solvable for $n \geq 5$.*

▶ **Lemma 8** ([7]). *If $n \equiv 0 \mod 2$ and $n > 2$ then the occurrence of an $(n-1)$-cycle, an n-cycle, and a permutation of type $2 + (n-3)$ on factoring the polynomial $f(x)$ modulo primes that do not divide the discriminant of $f(x)$ establishes that $Gal(f(x))$ over $\mathbb{Q}$ is the symmetric group S_n.*

Proof of Theorem 1. It suffices to show that f_δ is not solvable by the radicals as it describes the y-coordinates of two points in the solution. We provide three factorizations of f_δ modulo three primes that do not divide the discriminant $disc(f_\delta(x))$.

$$\begin{aligned}
f_\delta(x) &\equiv 12(x^{16} + 11x^{15} + 20x^{14} + 20x^{13} + 12x^{12} + 15x^{11} + 20x^{10} + 22x^9 + 19x^8 + 2x^7 + \\
&\qquad 18x^6 + 10x^5 + 12x^4 + 19x^3 + 16x^2 + 9x + 8) \mod 23 \\
f_\delta(x) &\equiv 21(x + 44)(x^2 + 34x + 39)(x^{13} + 4x^{12} + x^{11} + 41x^{10} + 12x^9 + 21x^8 + 24x^7 + \\
&\qquad 32x^5 + 22x^4 + 10x^3 + 24x^2 + 18x + 13) \mod 47 \\
f_\delta(x) &\equiv (x + 39)(x^{15} + 8x^{14} + 43x^{13} + 23x^{12} + 19x^{11} + 38x^{10} + 9x^9 + 6x^8 + 17x^7 + \\
&\qquad 34x^6 + 46x^5 + 43x^4 + 27x^3 + 50x^2 + 56x + 1) \mod 59
\end{aligned}$$

For the good primes $p = 23, 47$, and 59, the degrees of the irreducible factors of $f_\delta(x)$ mod p gives us an $16 - cycle$, a $2 + 13$ permutation and a 15-cycle, which is enough to show with Lemma 8 and $n = 16$ that $Gal(f_\delta) = S_{16}$. By Lemma 7, S_{16} is not solvable; with Lemma 6, this proves the theorem. ◄

3   Mowing Polyominoes

In the following, we analyze good tours for *polyominoes*, which naturally arise when a geometric (or geographic) region is mapped, resulting in axis-parallel edges and integer vertices. In the subsequent two sections, we describe the ensuing theoretical worst-case guarantees (Section 4) and the practical performance (Section 5).

3.1   Combinatorial Bounds

For a unit-square cutter, the LMP on polyominoes naturally turns into the TSP on the dual grid graph induced by pixel centers.

▶ **Lemma 9.** *Let $N \geq 2$ be the area of a polyomino P to be mowed with a unit-square cutter, and let L be the minimum length of a Lawn Mowing tour. Then $L \geq N$. In the case of a unit-square cutter, $L = N$ iff the dual grid graph of P has a Hamiltonian cycle.*

This follows from Lemma 2 in the paper by Arkin et al. [4] (which argues that there is an optimal LMP tour for a polyomino whose vertices are pixel centers) and implies the NP-hardness of the LMP (Theorem 1 in [4]). In particular, they focused on grid graphs without a *cut vertex*, which is a node v whose removal disconnects G: "If G has a cut vertex v, then we can consider separately the approximation problem in each of the components obtained by removing v, and then splice the tours back together at the vertex v to obtain a tour in the entire graph G. Thus, we concentrate on the case in which G has no cut vertices."

For a simply connected polyomino consisting of N pixels, the corresponding grid graph G does not have any *holes*, i.e., the complement of G in the infinite integer lattice is connected. These allow a tight combinatorial bound on the tour length. If G has no cut vertices, then a combinatorially bounded tour of G exists, as noted by Arkin et al. [4] as follows.

▶ **Theorem 10** (Theorem 5 in [4]). *Let G be a simple grid graph, having N nodes at the centerpoints, V, of pixels within a simple rectilinear polygon, R, having n (integer-coordinate) sides. Assume that G has no cut vertices. Then, in time $O(n)$, one can find a representation of a tour, T, that visits all N nodes of G, of length at most $\frac{6N-4}{5}$.*

For polyominoes with holes, there is a slightly worse, but still relatively tight combinatorial bound of $\frac{53N}{40} = 1.325N$ for the tour length, as follows.

▶ **Theorem 11** (Theorem 7 in [4]). *Let G be a connected grid graph, having N nodes at the centerpoints, V, of pixels within a (multiply connected) rectilinear polygon, R, having n (integer-coordinate) sides. Assume that G has no local cut vertices. Then, in time $O(n)$, one can find a representation of a tour, T, that visits all N nodes of G, of length at most $1.325N$.*

3.2   Mowing with a Disk

The natural lower bound of Lemma 9 still applies when mowing with a circular cutter, because any unit distance covered by the cutter can at most cover a unit area. However, meeting (or approximating) this bound is no longer possible by simply finding a Hamiltonian cycle (or a good tour) in the underlying grid graph, as a circular cutter may cover already mowed area or area outside of P when dealing with pixel corners. Minimizing this effect ultimately leads to the algebraic analysis from the previous section.

A starting point for further insights is illustrated in Figure 2: The optimal path from Lemma 4 with length L_{S_0} can be used for rectangles with width 2 and arbitrary height $h \geq 2$.
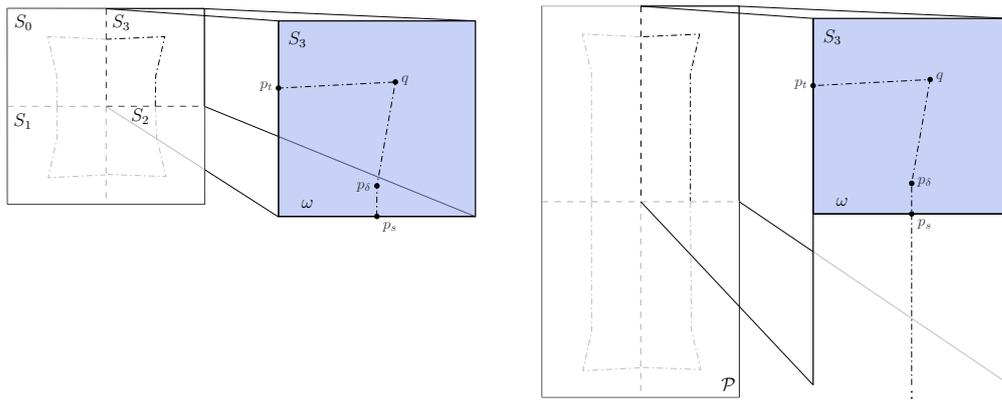
(a) Optimal Lawn Mowing tour for a 2×2 square. **(b)** Optimal Lawn Mowing tour for a rectangle.

■ **Figure 2** Optimal Lawn Mowing tours for a square and a rectangle.
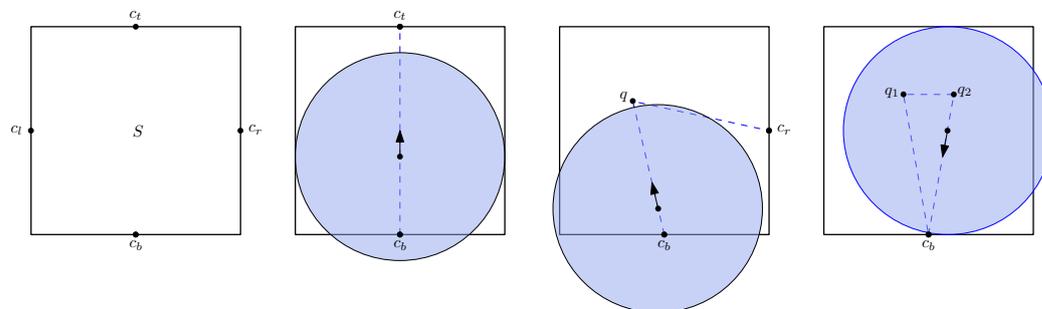
(a) A pixel S with transition points. **(b)** A straight-line covering, with zero turn. **(c)** A covering for a single, 90-degree turn. **(d)** A covering for a double, 180-degree U-turn.

■ **Figure 3** A pixel S with three elementary covering trajectories.

▶ **Corollary 12.** *Any rectangle P with width 2 and height $h > 2$ has a uniquely-shaped optimal Lawn Mowing tour T of length $L = 4L_{S_0} + 2h - 8$.*

Extending this idea to more general polyominoes leads to realizing a tour of the dual grid with locally optimal "puzzle pieces": a limited set of locally good trajectories that mow each visited pixel, which are merged at transition points on the pixel boundaries; see Figure 3a. The construction of the puzzle pieces is done in Section 3.3.

3.3 Constructing Puzzle Pieces

In order to analyze locally good trajectories for mowing visited pixels, consider the four corners of a pixel with coordinates $(0,0), (1,0), (1,1), (0,1)$. We consider *transition points* $c_b = (1/2, 0)$, $c_r = (1, 1/2)$, $c_t = (1/2, 1)$, and $c_l = (0, 1/2)$ at the edge centers to ensure an overall connected trajectory, as shown in Figure 3a. There are three combinatorially distinct ways for visiting a pixel, corresponding to Figures 3b–3d. These are (i) a straight path with length 1, (ii) a simple turn with length ≈ 1.32566, and (iii) a U-turn with length ≈ 1.611183, see the full version [27] for details on how to obtain these paths. Note that we do not use the optimal path from Lemma 4, because it uses transition points that are slightly off center, $p_t \neq c_r$, with the imbalance canceled out between two adjacent simple turns. Thus, using central transition points incurs a small marginal cost when compared to an optimal trajectory (1.32566 vs. 1.309, or about 1.2% longer for each simple turn), but it sidesteps the higher-order difficulties of combining longer off-center strips.

3.4   Building an Overall Tour

Making use of the puzzle pieces, we can now approach the LMP in three steps, as follows.

A Find a cheap roundtrip on the dual grid graph.
B Carry out the individual pixel transitions based on the above puzzle pieces as building blocks to ensure coverage of all pixels and thus a feasible tour.
C Perform post-processing sensitive to the transition costs on the resulting tour to achieve further improvement.

In the following sections, we describe how the involved steps can be carried out either with an emphasis on worst-case runtime and worst-case performance guarantee (giving rise to theoretical approximation algorithms, as discussed in the following Section 4), or with the goal of good practical performance in reasonable time for a suite of benchmark instances (leading to the experimental study described in Section 5).

4   Theoretical Performance: Approximation

For constant-factor approximation, we start with a low-cost roundtrip in the dual grid graph (Step A), e.g., with the previous results of Arkin et al. [4]. Step B is realized using the puzzle pieces of Section 3.3 for a feasible tour, at a cost of $1 + \tau := 1.32566$ for each 90-degree turn in the grid tour (corresponding to piece (ii)); note that the turn cost for a U-turn of 1.61118 (corresponding to piece (iii)) does not exceed $1 + 2\tau$. By using combinatorial arguments for the post-processing Step C, we can prove that a limited number of covering turns (with an additional turn cost τ) suffices for overall feasibility.

▶ **Theorem 13.** *Let P be a polyomino with $N > 5$ pixels, and let T be a tour of the dual grid graph of length L. Then we can find a feasible Lawn Mowing tour for a unit-diameter disk of length at most $L(1 + \tau)$.*

Proof. Let T be a tour of the dual grid graph; let L be the length of T. L is the total number of visits of individual pixels, inducing the following three categories of pixel visits.

1. L_0 "free" visits of pixels, in which no covering turn occurs, and no turn cost is incurred.
2. L_1 "one-turn" visits of pixels, in which one covering turn occurs, for a turn cost of τ.
3. L_2 "U-turn" visits of pixels, in which a double covering turn occurs, for a turn cost of not more than 2τ.

Let p_i be a pixel that is visited in step i of the tour by a U-turn of T. Then p_i is adjacent to a pixel $q = p_{i-1} = p_{i+1}$ that was left in step i and entered in step $i + 1$. Because no pixel visited by a U-turn needs to be visited more than once, as well as $N > 5$, the pixel q cannot only have neighbors that are visited by U-turns. Therefore, q has a predecessor in the tour that is not a U-turn, (w.l.o.g., p_{i-2}); this visit from p_{i-2} is either a one-turn visit with a covering turn, or a free visit. In either case, q is already covered when visited from p_i, and we can simply follow the grid path at only the distance cost of 1.

As a consequence, each U-turn visit (incurring a cost not exceeding 2τ) can be uniquely mapped to a free visit of its successor (incurring no turn cost), and the overall cost for all covering turns does not exceed $L\tau$, for a total length of at most $L(1 + \tau)$, as claimed. ◀

For simple polyominoes without cut vertices, Theorem 10 provides a tour T in the dual grid graph of length at most $\frac{6N-4}{5}$, implying the following.

▶ **Corollary 14.** *Let P be a simple polyomino with n vertices and N pixels, whose dual grid graph does not have any cut vertices. Then, in time $O(n)$, one can find a representation of a feasible Lawn Mowing trajectory T for a unit-diameter disk of length at most $\frac{6N-4}{5}\tau$, which is within 1.5908 of the optimum.*

For polyominoes with holes, we can apply the same line of argument to a tour T of the dual grid graph obtained from Theorem 11.

▶ **Corollary 15.** *Let P be a (not necessarily simple) polyomino with n vertices and N pixels, whose dual grid graph does not have any cut vertices. Then, in time $O(n)$, one can find a representation of a feasible Lawn Mowing trajectory T for a unit-diameter disk of length at most $\frac{53N}{40}\tau$, which is within 1.7565 of the optimum.*

As the number of turns is of critical importance for the overall cost of a Lawn Mowing tour obtained from a tour of the dual grid graph, we can consider optimizing a linear combination of tour length and turn cost. Arkin et al. [2] gave a PTAS for this problem, as follows.

▶ **Theorem 16** (Theorem 5.17 in [2]). *Define the cost of a tour to be its length plus C times the number of (90-degree) turns. For any fixed $\varepsilon > 0$, there is a $(1 + \varepsilon)$-approximation algorithm, with running time $2^{O(h)}N^{O(C)}$, for minimizing the cost of a tour for an integral orthogonal polygon P with h holes and N pixels.*

Combining tour length and turns allows providing more explicit bounds, as follows. Additional local considerations are possible, but these do not necessarily improve the worst-case bounds. Instead, they are employed heuristically in the practical section.

▶ **Theorem 17.** *Let P be a polyomino with n vertices and N pixels, and let T be a tour of the dual grid graph of length L and a total of t (weighted) turns. Then there is a feasible Lawn Mowing tour of cost at most $L + t\tau$.*

5   Practical Performance: Algorithm Engineering

5.1   Algorithmic Tools

Here we exploit the algorithmic approach of Section 3.4 for good *practical* performance for general polygonal regions, starting with a preprocessing step: For a given polygonal region Q, find a suitable polyomino P that covers it.

We can then aim for practical minimization of tour length and turn cost for **A** (analogous to the theoretical Theorem 16), and use puzzle pieces in **B** for a feasible tour. In principle, we can approach **A** by considering an integer program (IP); however, solving this IP becomes too costly for larger instances, so we use a more scalable approach: **(A)** Find a good TSP solution on the dual grid graph; **(B)** insert puzzle pieces; **(C)** minimize the induced turn cost by Integer Programming and Large Neighborhood Search (LNS).

5.1.1   Choosing a Suitable Grid

Consider a non-degenerate polygonal region Q, and a minimal covering polyomino P of cell size ℓ. Without loss of generality, Q contains only pixels with a point of Q in their interior; furthermore, we can assume that both an x- and a y-coordinate of a grid point coincide with a coordinate of Q. This limits the number of relevant grid positions to a quadratic number of choices, from which one can choose the one with the smallest number of pixels contained in the resulting polygon P.

5.1.2 Minimizing Tour and Turn Cost

Finding a covering tour of minimum combined tour length and turn cost can be formulated as an IP. As the cost for each turn can be specified individually in this IP, we can also minimize the final tour length directly instead of just approximating it based on the number of turns. In principle, this IP can be solved with CPLEX [17] or Gurobi [30]; however, this fails when aiming for truly large instances. (Even without the length of the tour, the turn-cost problem is notoriously difficult [25].) Thus, we have pursued an alternative approach that starts with a cheap roundtrip on the dual grid graph in which we ignore the turn cost. We then use this IP as part of a Large Neighborhood Search (described in Section 5.1.4) to minimize the actual costs of this solution, and for computing lower bounds on the best possible solution based on puzzle pieces.

Formulating the Integer Program. To formulate the integer program, let $\widehat{uvw}$ with $uv, vw \in E_H$ be the puzzle piece covering the pixel v and connecting c_{uv} and c_{vw}, and $\overline{uvw}$ be the direct path between c_{uv} and c_{vw}. We call these tour elements (*covering* and *non-covering*) *tiles*. We use the variables $x_{\widehat{uvw}} \in \mathbb{B}, uv, vw \in E_H$ to denote which covering tile, i.e., puzzle piece, is used for $v \in V_P$ is in the tour. For simplicity, $x_{\widehat{uvw}}$ is also defined for $v \in V \setminus V_P$, but fixed to 0. Analogously, we are using the variables $x_{\overline{uvw}} \in \mathbb{N}_0, uv, vw \in E_H$ to denote how often which non-covering tiles, i.e., direct paths, for $v \in V$ are used in the tour. Because we may need to pass a pixel multiple times, this is an integer variable.

Finding the shortest set of cycles that cover all pixels $t \in V_P$ can be expressed as follows. Enforcing a single cycle, i.e., tour, is done later by some more complex constraints that need additional discussion.

$$\min \sum_{uv, vw \in E_H} ||\overline{uvw}|| \cdot x_{\overline{uvw}} + ||\widehat{uvw}|| \cdot x_{\widehat{uvw}} \tag{2}$$

$$\text{s.t.} \sum_{u, w \in N(v)} x_{\widehat{uvw}} = 1 \qquad\qquad \forall v \in V_P \tag{3}$$

$$2 \cdot (x_{\overline{wvw}} + x_{\widehat{wvw}}) + \sum_{n \in N(v), n \neq w} (x_{\overline{nvw}} + x_{\widehat{nvw}})$$
$$= 2 \cdot (x_{\overline{vwv}} + x_{\widehat{vwv}}) + \sum_{n \in N(w), n \neq v} (x_{\overline{vwn}} + x_{\widehat{vwn}}) \qquad \forall vw \in E_H \tag{4}$$

$$x_{\widehat{vwv}} \in \mathbb{B}, x_{\overline{vwv}} \in \mathbb{N}_0 \qquad\qquad \forall uv, vw \in E_H \tag{5}$$

The objective function (2) minimizes the sum of lengths of the used tiles (the length of a tile is denoted by $|| \cdot ||$). Equation (3) enforces that every pixel $v \in V_P$ that intersects the polygon P has one covering tile; $N(v)$ are the neighbors of v. Equation (4) ensures that every tile has a matching incident tile on each end, i.e., connecting all tiles yields feasible cycles.

Subtour Elimination. Next, we have to add constraints that enforce a single tour. A simple, but insufficient, constraint is similar to the classical subtour elimination constraint of the Dantzig-Fulkerson-Johnson formulation [18] for the Traveling Salesman Problem. For every non-empty subset $S \subset V, S \neq \emptyset, V \not\subset S, V \setminus S \neq \emptyset$ that contains a real part of V_P, there has to be some path leaving the set to connect to $V \setminus S$.

$$\sum_{uv, vw \in E_H, v \in S, w \notin S} x_{\overline{uvw}} + x_{\widehat{uvw}} \geq 1$$

Unfortunately, this is not sufficient as we can have cycles that cross but are not connected, e.g., for the tiles $\overline{uvw}$ and $\widehat{svt}$ with $\{u, w\} \cap \{s, t\} = \emptyset$. While they share the same pixel v in the grid graph, the paths themselves do not have to intersect. We can also not expect them to be exchangeable as this may increase the objective. Let O be a cycle of tiles that cover only a real subset of V_P, $E(O)$ denote the edges in the grid graph, and $\widehat{abc} \in O$ be a covering tile of O with $b \in V(O) \cap V_P$. The following constraint now forces the path that covers v to change and connect to exterior parts.

$$\sum_{u, w \in N(b), \widehat{ubw} \notin O} x_{\widehat{ubw}} + \sum_{vw \in E(O), u \in N(v), \overline{uvw} \notin O, \widehat{uvw} \notin O} (x_{\overline{uvw}} + x_{\widehat{uvw}}) \geq 1$$

This constraint is sufficient as it can be applied to any cycle that is covering only a subset of V_P, but generally less efficient.

5.1.3   Finding a Cheap Roundtrip and Ensure Coverage

We consider two different methods for computing different initial tours.

TSP_Small.   Previous authors [12, 37, 42, 44] have suggested using a grid graph H' with smaller cell size $\ell = \frac{\sqrt{2}}{2}$ for covering P, or simply assumed square-shaped tools. This eliminates the need to consider any turn cost, as smaller pixels are covered when the cutter visits their centers. This yields $\mathrm{TSP}_{\mathrm{Small}}$, which we use as a baseline. Because of the smaller grid size, this may result in double coverage when parallel unit strips suffice to cover the P, for a worst-case overhead of $\sqrt{2} - 1$, or about 41.4 %.

TSP_Cov.   As described in the preceding Section 3, we can use a cheap tour for the grid graph H with cell size $\ell = 1$, and perform the puzzle piece modification. This combined solver is called $\mathrm{TSP}_{\mathrm{Cov}}$. As shown in Section 4, we can limit the worst-case overhead for performing turns of $\mathrm{TSP}_{\mathrm{Cov}}$ to $\tau = 0.32566$ per length of the tour, or about 32.6 %.

5.1.4   Improving the Tour

For a feasible tour from $\mathrm{TSP}_{\mathrm{Cov}}$, we use an LNS-algorithm [41], which iteratively fixes a large part of the IP and only optimizes a small region of tiles; this yields $\mathrm{TSP}_{\mathrm{Turn}}$. We select a random tile from the current tour and a fixed number of adjacent pixels. This yields a limited-size integer program, in which only the involved puzzle pieces are allowed to change. To escape local minima, we tune the size (and runtime) of the IP after each iteration based on the runtime of the previous iteration. In the end, we attempt to solve the IP on the complete instance, using the start solution from the LNS. This provides lower bounds on the best placement of puzzle pieces.

5.2   Experimental Setup

Our practical implementation was tested on a workstation with an AMD Ryzen 7 5800X $(8 \times 3.8\,\mathrm{GHz})$ CPU and $128\,\mathrm{GB}$ of RAM. The code and data are publicly available[1]. We used the *srpg_iso*, *srpg_iso_aligned* and *srpg_octa* instances and generated additional polyominoes with the open-source code from the Salzburg Database of Geometric Inputs [19].

---

[1] `https://github.com/tubs-alg/lawn-mowing-from-algebra-to-algorithms`

(a) Distribution of instances.

(b) Three instances with the cutter C.
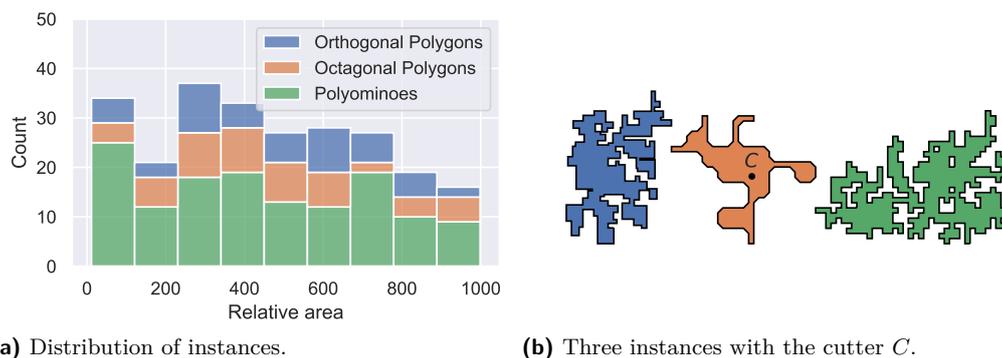
Figure 4 Examples of the used polygons and their size distribution.

See Figure 4 for the overall distribution and Figure 12 in the full version [27] for examples. We considered polygons with up to $n = 300$ vertices and a cutter with diameter 1. Overall, this resulted in 327 instances. All experiments were carried out with a maximum runtime of 300 s for TSP, LNS and final IP computation. To solve the TSP efficiently, we used the python binding *pyconcorde* of the Concorde TSP Solver [43]. All components of TSP_{Cov} and TSP_{Turn} were implemented in Python 3.10 and used the IP solver Gurobi (v10.0) [30]. As in previous work [26] the *relative area* (ratio of convex hull area of P and cutter area $A(C)$) is more significant for the difficulty of an instance than number of vertices of P.
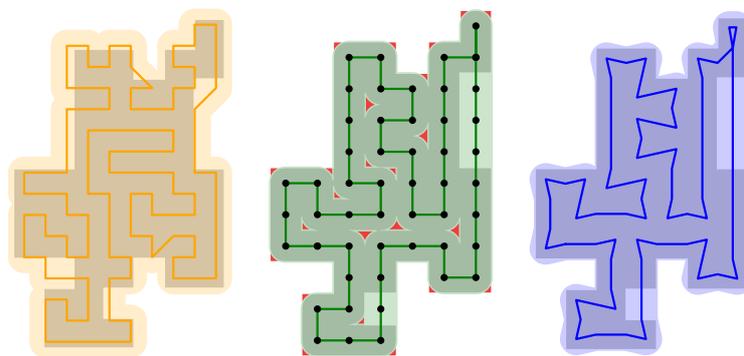
Figure 5 (Left) A $\text{TSP}_{\text{Small}}$ tour yields a feasible but expensive LMP tour. (Middle) A TSP tour of the underlying dual grid graph, with uncovered patches shown in red. (Right) A feasible LMP tour after puzzle piece modification of the TSP tour.

5.3 Evaluation

We discuss our practical results along a number of research questions (RQ).

RQ1: How does TSP_{Cov} compare to $\text{TSP}_{\text{Small}}$ in practice? We compared the worst-case bound of 32.6 % for TSP_{Cov} to the actual performance, using the total cost of $\text{TSP}_{\text{Small}}$ as a baseline. See Figure 5 for an example and Figure 6 for the average relative modification cost. This shows not more than an additional 19 % cost, with only small variation over size and
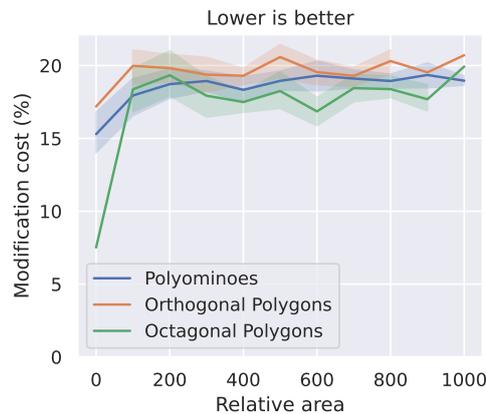
Figure 6 Modification cost over size and instance type. The modification induces a cost of around 19 % over all instance types and sizes. The plot shows the average modification cost and the 95 % confidence interval.

type. Figure 7b shows that the practical average reduction from TSP_{Small} is independent of the size of the polygon, but differs strongly for the different instance classes; we save $\approx 27\%$ for polyominoes, $\approx 24\%$ for octagonal polygons, and $\approx 5\%$ for orthogonal polygons.

RQ2: How good are the solutions achieved by TSP_{Turn}?   For the considered large instances, provably optimal solutions for the turn-cost minimizing IP are hard to find, so we considered the remaining optimality gap in the IP. Figure 7a shows that gaps remain below 7 % even for large instances, and below 5 % on average for medium-sized instances.

We also compared the tours from TSP_{Cov} with the cheapest tours obtained by TSP_{Turn} and TSP_{Small}. As shown in Figure 7b, on average we obtain $\approx 5\%$ shorter tours when compared to the TSP_{Cov} tours, independent of instance size and type. For orthogonal polygons, this doubles the cost reduction.

RQ3: How far are we from the geometric area lower bound?   A remaining gap between TSP_{Turn} and the area bound may result from two sources, both from (i) the quality of the upper bound (and thus TSP_{Turn}) and (ii) the quality of the area lower bound, for the following reasons. (i) The optimal LMP tour is not restricted to the grid graph H, so there may be cheaper tours than what we obtain from TSP_{Turn}. (ii) The simple area bound (corresponding to Lemma 9) is relatively weak, so it is conceivable that a serious gap to this lower bound remains.

Overall, the combination of both effects remains limited, as can be seen from Figure 8 (showing the ratio of TSP_{Turn} value and area bound): For the octagonal polygons and polyominoes, we are on average at most 50 % above the area bound. For orthogonal polygons, the relative gap is on average below 80 %.

RQ4: How do our solutions compare to previous practical work?   As shown before, our results are already considerably better than work based on TSP_{Small}. A comparison to the previous best practical results by Fekete et al. [26] (whose instances were used as a subset of our benchmarks) is shown in Figure 8; plotted are the ratios between the achieved solution values and the respective lower bounds. Fekete et al. [26] employ a more sophisticated lower bound based on an evaluation of a series of Close-Enough TSP (CETSP) instances. The
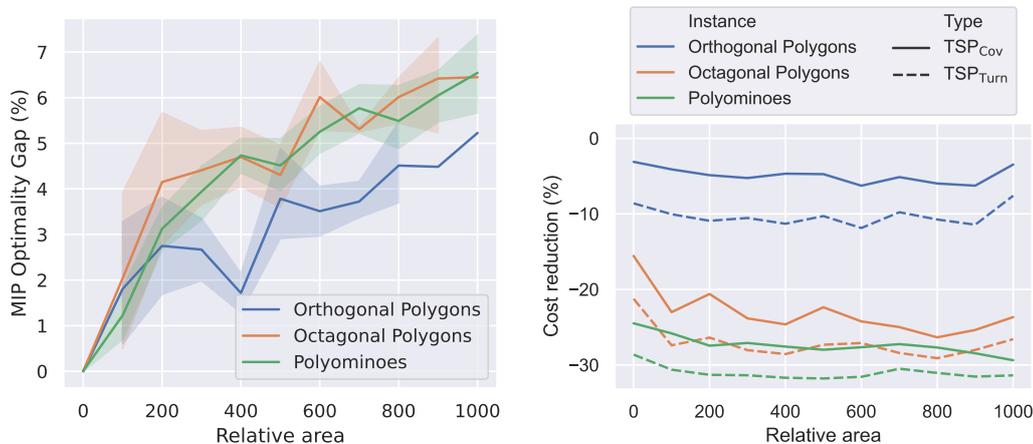
(a) Optimality gap of the IP.

(b) Cost reduction.

Figure 7 (a) Remaining average optimality gaps for the integer program and the 95 % confidence interval. (b) Comparison of the average cost reduction for different approaches and polygon types.
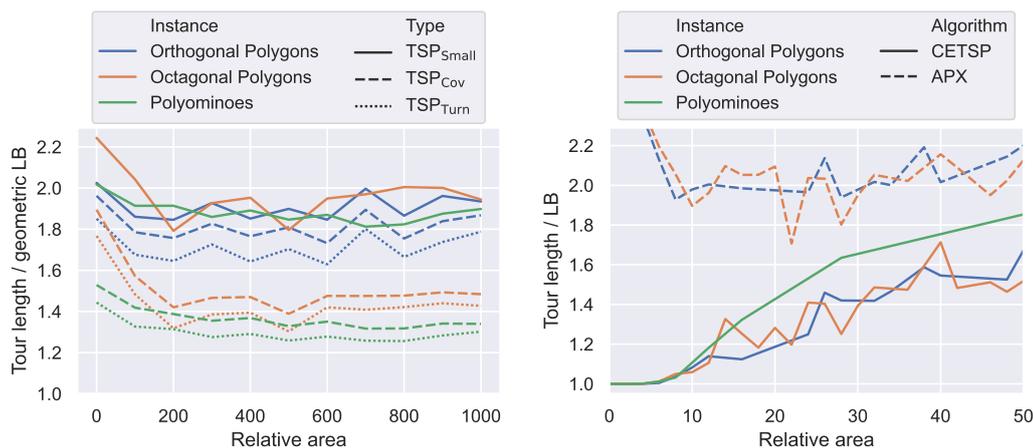
(a) Tour length compared to the area bound.

(b) Solution quality of [26].

Figure 8 (a) Tour length compared to the (weaker) area lower bound in terms of the average ratio. For octagonal polygons and polyominoes, we can get below 50 % on average. (b) Comparable results for the average solution quality of [26] based on a (stronger) CETSP bound; here APX denotes the performance of the approximation algorithm by [4]. Note the considerably larger relative area in comparison to [26].

authors pointed out that the lower bound computation becomes very expensive even for instances with relative area smaller than 50, see Figures 11 and 12 in [26]. Because we evaluate much larger instances, our ratios only use the relatively straightforward area bound. As a consequence, the denominators of these ratios favor the evaluation for [26], which are shown in Figure 8b; see Figures 9a and 9b for a comparison on a relatively small example that was also shown in [26]. In addition, we were able to achieve results for instances with a relative area 20 times larger than [26].

Despite these additional challenges (of weaker bounds and larger instance sizes), our results compare favorably to the ones reported by [26]. The main reason lies in our structurally simpler approach that still yields good results when the complex evaluation of the CETSP from [26] reaches computational limitations. As can be seen from a comparison of computed trajectories for the visual example (Figures 9c and 9d), this is also reflected in simpler trajectories obtained from TSP$_{\text{Turn}}$.
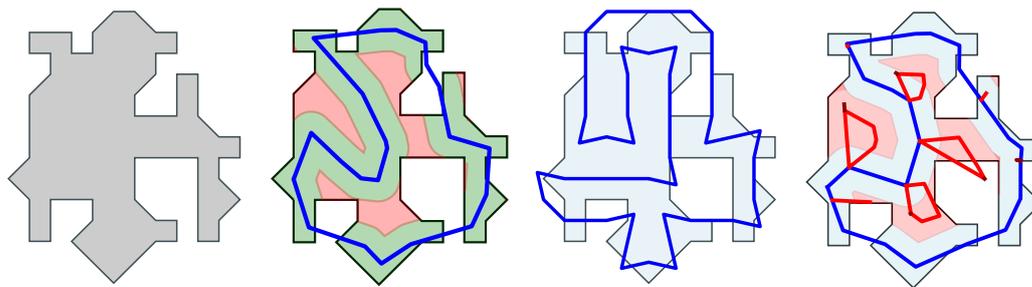
(a) Area (LB): 36.25. **(b)** LB: 40.94 [26]. **(c)** TSP$_{\text{Turn}}$ (UB): 66.71. **(d)** UB: 68.16 [26].

Figure 9 Comparison of TSP$_{\text{Turn}}$ with lower and upper bounds from Fekete et al. [26].

6 Conclusion

We have presented new insights for the Lawn Mowing Problem, starting with an algebraic analysis of the structure of optimal trajectories. As a consequence, we can pinpoint a particular source of the perceived overall difficulty of the problem, and prove that constructing optimal tours necessarily involves operations that go beyond simple geometric means; we can also use these insights to come up with better construction methods for tours, both on the theoretical and the practical side, with minimizing overall turn cost playing a crucial role.

Our results also clear the way for a number of important followup questions. Is it possible to improve our approach for polyominoes? As discussed in the text, considering higher-order connectivity between turns and using slightly off-center, axis-parallel strips appears to be a relatively easy way for (albeit marginal) improvement. It may very well be that this ultimately leads to optimal tours for polyominoes; however, final success on this fundamental challenge will require another breakthrough in establishing lower bounds, as neither the polygon area (which may incur a gap from the optimal value, similar to the number of vertices in a grid graph does from a TSP solution) nor the Close-Enough TSP bound for a finite set of witness points may suffice to certify optimality. Given that an optimal tour may also involve portions that are not axis-parallel, it will also require further algebraic analysis of turns that are not multiples of 90 degrees.

For the Lawn Mowing Problem on general regions (which may not even have to be connected), our hardness result hints at further difficulties. It is quite conceivable that the general LMP is not just algebraically hard, but even $\exists\mathbb{R}$-complete. Even in that case, we believe that further engineering of the tile-based mowing of polyominoes (with attention to turn cost) and Close-Enough TSP may be the most helpful tools for further systematic improvement.

─── **References** ───

1 David L Applegate, Robert E Bixby, Vašek Chvátal, and William J Cook. *The Traveling Salesman Problem: A Computational Study.* Princeton Series in Applied Mathematics. Princeton University Press, 2007. `doi:10.1016/j.orl.2007.06.002`.

2 Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S. B. Mitchell, and Saurabh Sethia. Optimal covering tours with turn costs. *SIAM Journal on Computing*, 35(3):531–566, 2005. `doi:10.1137/S0097539703434267`.

3 Esther M. Arkin, Sándor P. Fekete, and Joseph S. B. Mitchell. The lawnmower problem. In *Canadian Conference on Computational Geometry (CCCG)*, pages 461–466, 1993. URL: `https://cglab.ca/~cccg/proceedings/1993/Paper79.pdf`.

4 Esther M. Arkin, Sándor P. Fekete, and Joseph S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17:25–50, 2000. `doi:10.1016/S0925-7721(00)00015-8`.

5 Esther M. Arkin, Martin Held, and Christopher L. Smith. Optimization problems related to zigzag pocket machining. *Algorithmica*, 26(2):197–236, 2000. `doi:10.1007/s004539910010`.

6 Rik Bähnemann, Nicholas Lawrance, Jen Jen Chung, Michael Pantic, Roland Siegwart, and Juan Nieto. Revisiting Boustrophedon coverage path planning as a generalized traveling salesman problem. In *Field and Service Robotics*, pages 277–290, 2021. `doi:10.1007/978-981-15-9460-1_20`.

7 Chanderjit Bajaj. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, 3(2):177–191, 1988. `doi:10.1007/BF02187906`.

8 Michael J. Bannister, William E. Devanny, David Eppstein, and Michael T. Goodrich. The galois complexity of graph drawing: Why numerical solutions are ubiquitous for force-directed, spectral, and circle packing drawings. *Journal of Graph Algorithms and Applications*, 19(2):619–656, 2015. `doi:10.7155/jgaa.00349`.

9 Aaron T. Becker, Mustapha Debboun, Sándor P. Fekete, Dominik Krupke, and An Nguyen. Zapping zika with a mosquito-managing drone: Computing optimal flight patterns with minimum turn cost. In *Symposium on Computational Geometry (SoCG)*, pages 62:1–62:5, 2017. Video at `https://www.youtube.com/watch?v=SFyOMDgdNao`. `doi:10.4230/LIPIcs.SoCG.2017.62`.

10 Károly Bezdek. *Körök optimális fedései (Optimal Covering of Circles).* PhD thesis, Eötvös Lorand University, 1979.

11 Károly Bezdek. Über einige optimale Konfigurationen von Kreisen. *Ann. Univ. Sci. Budapest Rolando Eötvös Sect. Math*, 27:143–151, 1984.

12 Richard Bormann, Joshua Hampp, and Martin Hägele. New brooms sweep clean - an autonomous robotic cleaning assistant for professional office cleaning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4470–4477, 2015. `doi:10.1109/ICRA.2015.7139818`.

13 Tauã M. Cabreira, Lisane B. Brisolara, and Paulo R. Ferreira Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4, 2019. `doi:10.3390/drones3010004`.

14 Jean-Lou De Carufel, Carsten Grimm, Anil Maheshwari, Megan Owen, and Michiel H. M. Smid. A note on the unsolvability of the weighted region shortest path problem. *Computational Geometry*, 47(7):724–727, 2014. `doi:10.1016/j.comgeo.2014.02.004`.

15 Howie Choset. Coverage for robotics–a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, 2001. `doi:10.1023/A:1016639210559`.

16 Howie Choset and Philippe Pignon. Coverage path planning: The Boustrophedon cellular decomposition. In *Field and Service Robotics*, pages 203–209, 1998. `doi:10.1007/978-1-4471-1273-0_32`.

17 International Business Machines Corporation. IBM ILOG CPLEX Optimization Studio, 2023.

18 George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. `doi:10.1287/opre.2.4.393`.

19   Günther Eder, Martin Held, Steinþór Jasonarson, Philipp Mayer, and Peter Palfrader. Salzburg database of polygonal data: Polygons and their generators. *Data in Brief*, 31:105984, 2020. `doi:10.1016/j.dib.2020.105984`.

20   Gershon Elber and Myung-Soo Kim. Offsets, sweeps and Minkowski sums. *Computer-Aided Design*, 31(3), 1999. `doi:10.1016/S0010-4485(99)00012-3`.

21   Brendan Englot and Franz Hover. Sampling-based coverage path planning for inspection of complex structures. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 29–37, 2012. URL: `http://www.aaai.org/ocs/index.php/ICAPS/ICAPS12/paper/view/4728`.

22   Gábor Fejes Tóth. Thinnest covering of a circle by eight, nine, or ten congruent circles. *Combinatorial and Computational Geometry*, 52:361–376, 2005. URL: `http://library.msri.org/books/Book52/files/18fejes.pdf`.

23   Sándor P. Fekete, Utkarsh Gupta, Phillip Keldenich, Christian Scheffer, and Sahil Shah. Worst-case optimal covering of rectangles by disks. In *Symposium on Computational Geometry (SoCG)*, pages 42:1–42:23, 2020. `doi:10.4230/LIPIcs.SoCG.2020.42`.

24   Sándor P. Fekete, Phillip Keldenich, and Christian Scheffer. Covering rectangles by disks: The video. In *Symposium on Computational Geometry (SoCG)*, pages 71:1–75:5, 2020. `doi:10.4230/LIPIcs.SoCG.2020.75`.

25   Sándor P. Fekete and Dominik Krupke. Practical methods for computing large covering tours and cycle covers with turn cost. In *Algorithm Engineering and Experiments (ALENEX)*, pages 186–198, 2019. `doi:10.1137/1.9781611975499.15`.

26   Sándor P. Fekete, Dominik Krupke, Michael Perk, Christian Rieck, and Christian Scheffer. A closer cut: Computing near-optimal lawn mowing tours. In *Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 1–14, 2023. `doi:10.1137/1.9781611977561.ch1`.

27   Sándor P. Fekete, Dominik Krupke, Michael Perk, Christian Rieck, and Christian Scheffer. The lawn mowing problem: From algebra to algorithms, 2023. `doi:10.48550/arXiv.2307.01092`.

28   Erich Friedman. Circles covering squares web page. `https://erich-friedman.github.io/packing/circovsqu`, 2014. Online, accessed January 10, 2023.

29   Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013. `doi:10.1016/j.robot.2013.09.004`.

30   Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

31   Martin Held. *On the Computational Geometry of Pocket Machining*, volume 500 of *LNCS*. Springer, 1991. `doi:10.1007/3-540-54103-9`.

32   Martin Held, Gábor Lukács, and László Andor. Pocket machining based on contour-parallel tool paths generated by means of proximity maps. *Computer-Aided Design*, 26(3):189–203, 1994. `doi:10.1016/0010-4485(94)90042-6`.

33   Aladár Heppes and Hans Melissen. Covering a rectangle with equal circles. *Periodica Mathematica Hungarica*, 34(1-2):65–81, 1997. `doi:10.1023/A:1004224507766`.

34   Israel N. Herstein. *Topics in algebra*. John Wiley & Sons, 1991.

35   Katharin R. Jensen-Nau, Tucker Hermans, and Kam K. Leang. Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring. *IEEE Transactions on Automation Science and Engineering*, 18(3):1453–1468, 2021. `doi:10.1109/TASE.2020.3016276`.

36   Johannes B. M. Melissen and Peter C. Schuur. Covering a rectangle with six and seven circles. *Discrete Applied Mathematics*, 99(1-3):149–156, 2000. `doi:10.1016/S0166-218X(99)00130-4`.

37   Ghulam Murtaza, Salil S. Kanhere, and Sanjay K. Jha. Priority-based coverage path planning for aerial wireless sensor networks. In *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (IPSN)*, pages 219–224, 2013. `doi:10.1109/ISSNIP.2013.6529792`.

38   Eric H. Neville. On the solution of numerical functional equations. *Proceedings of the London Mathematical Society*, 2(1):308–326, 1915. `doi:10.1112/plms/s2_14.1.308`.

39   David Nistér, Richard I. Hartley, and Henrik Stewénius. Using galois theory to prove structure from motion algorithms are optimal. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. `doi:10.1109/CVPR.2007.383089`.

40   Timo Oksanen and Arto Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8):651–668, 2009. `doi:10.1002/rob.20300`.

41   David Pisinger and Stefan Ropke. Large neighborhood search. *Handbook of metaheuristics*, pages 99–127, 2019. `doi:10.1007/978-3-319-91086-4_4`.

42   Gokarna Sharma, Ayan Dutta, and Jong-Hoon Kim. Optimal online coverage path planning with energy constraints. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1189–1197, 2019. URL: `https://dl.acm.org/doi/10.5555/3306127.3331820`.

43   Solver, Concorde TSP. Concorde TSP Solver, 2023. URL: `https://www.math.uwaterloo.ca/tsp/concorde.html`.

44   Xiaoming Zheng, Sven Koenig, David Kempe, and Sonal Jain. Multirobot forest coverage for weighted and unweighted terrain. *IEEE Transactions on Robotics*, 26(6):1018–1031, 2010. `doi:10.1109/TRO.2010.2072271`.