Minimum Link Fencing

Sujoy Bhore
Department of Computer Science & Engineering, Indian Institute of Technology Bombay, India

Fabian Klute
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Maarten Löffler
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Martin Nöllenburg
Algorithms and Complexity Group, TU Wien, Austria

Soeren Terziadis
Algorithms and Complexity Group, TU Wien, Austria

Anaïs Villedieu
Algorithms and Complexity Group, TU Wien, Austria

Abstract

We study a variant of the geometric multicut problem, where we are given a set $\mathcal{P}$ of colored and pairwise interior-disjoint polygons in the plane. The objective is to compute a set of simple closed polygon boundaries (*fences*) that separate the polygons in such a way that any two polygons that are enclosed by the same fence have the same color, and the total number of links of all fences is minimized. We call this the *minimum link fencing* (MLF) problem and consider the natural case of *bounded minimum link fencing* (BMLF), where $\mathcal{P}$ contains a polygon Q that is unbounded in all directions and can be seen as an outer polygon. We show that BMLF is NP-hard in general and that it is XP-time solvable when each fence contains at most two polygons and the number of segments per fence is the parameter. Finally, we present an $O(n \log n)$-time algorithm for the case that the convex hull of $\mathcal{P} \setminus \{Q\}$ does not intersect Q.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases computational geometry, polygon nesting, polygon separation

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.34

Related Version *Full Version*: arXiv:2209.14804 [4]

Funding *Fabian Klute*: Supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 612.001.651 and the Austrian Science Foundation (FWF) grant J4510.
Anaïs Villedieu: Supported by the Austrian Science Fund (FWF) under grant P31119.

Acknowledgements The authors would like to thank Thekla Hamm and Irene Parada for the valuable discussion in particular concerning the XP results of Section 3.

1 Introduction

In the *geometric multicut* problem [2], we are given κ disjoint sets of polygons in the plane, each with a different *color*, and are asked for a subdivision of the plane such that no cell of the subdivision contains multiple colors. The goal is to minimize the total length of the subdivision edges.

A different kind of separation is achieved in the *polygon nesting* problem [3], where for two polygons P and Q with $P \subset Q$ one asks for a polygon P' with the smallest number of links, s.t. $P \subset P' \subset Q$. There exists a series of work that addressed the algorithmic complexity of nesting problems for various polygon families [3, 5, 9, 12, 13]. See Section 1.2 for more detail.

33rd International Symposium on Algorithms and Computation (ISAAC 2022).
Editors: Sang Won Bae and Heejin Park; Article No. 34; pp. 34:1–34:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

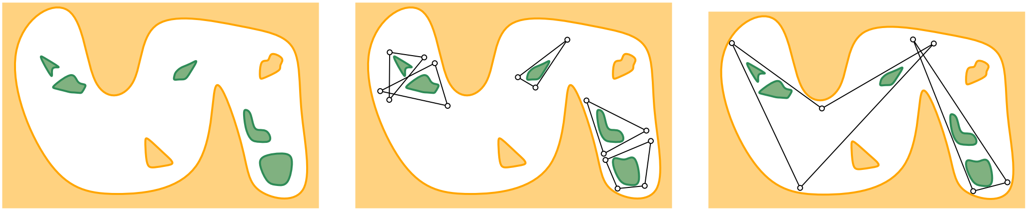


Figure 1 Two sets of polygons in the plane (left) with different colors (green and yellow). The yellow set effectively acts as an outer polygon with holes. Separating the two sets with, possibly intersecting, individual fences (middle) can lead to significantly more links in the fences (here 16) than grouping same-colored polygons (right), which achieves this with just seven links.

In this paper, we consider a variant of geometric multicut inspired by polygon nesting, where we separate the sets from each other with a set of closed polygon boundaries called *fences*, which enclose only polygons of one color and have the smallest possible number of links. If one or more sets are not connected, we need to solve the combinatorial problem of choosing which polygons should be grouped in each fence. Figure 1 illustrates the problem. Some variants of the fencing problem already become NP-hard for point objects with two colors, e.g., if we require the fence to be a single closed curve [6].

In this paper, we assume the input sets are collections of polygons, one color covers the plane minus a single polygonal hole (the outer polygon, a parallel to polygon nesting), and we will focus on the case $\kappa = 2$ of two colors. We use n to denote the total number of corners of the input polygons. Even in this simple setting the problem turns out to be non-trivial. If both sets are connected, then the problem is equivalent to finding a minimal *nested* polygon, which can be solved in $O(n \log n)$ time [3]. If both sets are not connected we show this problem to be NP-hard in Section 2. Note the contrast to the geometric multicut problem, which is polynomially solvable for $\kappa = 2$ [1] but becomes NP-hard when $\kappa = 3$ [2]. In Section 3 we show that, when restricting every fence to contain at most two polygons, the problem admits an XP-algorithm when parameterized by the maximal number of segments per fence, a result which holds for any κ. Finally, in Section 4, we show that the problem is polynomial-time solvable if the convex hull of the second color (the *inner polygons*) is contained in the outer polygon and the first color is connected.

1.1 Problem Definition

Throughout this paper we consider polygons in $\mathbb{R}^2$ without self-intersections but potentially with holes. Moreover, we consider a polygon as the boundary together with its interior, unless stated otherwise. We consider the following problem.

▶ **Definition 1** (Minimum Link Fencing (MLF)). *We are given n pairwise interior-disjoint polygons $\mathcal{P} = \{P_1, \dots, P_{|\mathcal{P}|}\}$ in the plane, with a coloring function $f : \mathcal{P} \to \{1, \dots, \kappa\}$, which assigns a color to every input polygon. We write $\mathcal{P}_i = \{P \mid f(P) = i\}$. We want to find a set of simple closed polygon boundaries $\mathcal{F} = \{F_1, \dots, F_m\}$ such that the total number of links $|F|$ on the boundary of $F = \bigcup_{i=1}^{k} F_i$ is minimized and if two polygons P_a and P_b are enclosed by the same fence or are both in $\mathbb{R}^2 \setminus \bigcup_{i=1}^{k} \overline{F_i}$, where $\overline{F_i}$ is the polygon bounded by F_i, then $f(P_a) = f(P_b)$. We call F_i a fence and $\mathcal{F}$ a* minimum link fencing *of $\mathcal{P}$.*

Note the important difference in Definition 1 between $\mathcal{F}$, which is the set of all fences of a solution, and F, which is the union over all fences, i.e., one (possibly disconnected) polygon. Thus $|\mathcal{F}|$ is the number of fences and $|F|$ is the number of all segments in these fences.

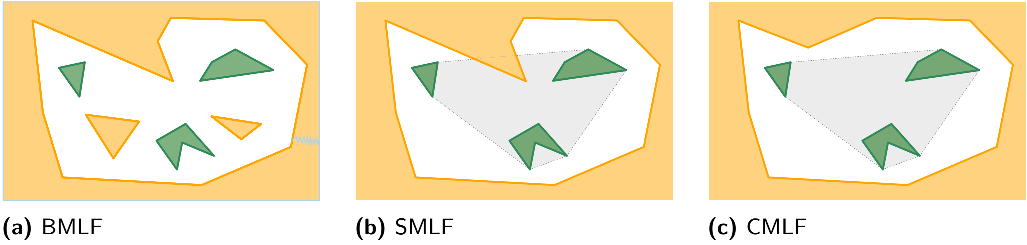

(a) BMLF (b) SMLF (c) CMLF

Figure 2 Different problem inputs corresponding to (a) BMLF, (b) SMLF and (c) CMLF. In (b) and (c) the convex hull of all input polygons indicated in gray.

Throughout the paper we refer to $\mathbb{R}^2 \setminus \bigcup_{i=1}^{|\mathcal{P}|} P_i$ as the *free space* (between polygons). We refer to $\mathcal{P}$, which contains polygons of κ different colors, as κ-colored and to the problem setting as the κ-colored problem. We consider several problem variations.

If there exists a polygon $Q \in \mathcal{P}$ which is unbounded in every direction, i.e. $\mathbb{R}^2 \setminus Q$ is finite, this polygon Q effectively acts as an outer boundary. In this case we call the problem Bounded Minimum Link Fencing (BMLF). We denote the polygon Q as the *outer polygon*. As a consequence, the size of the outer polygon automatically bounds the length of any link in a fence. Else, in general, one fence could conain a very long link, while retaining small complexity when counting the number of links only. Note that Q can be emulated in an instance of Minimum Link Fencing, by adding a large rectangular polygon $P_c \setminus (\mathbb{R}^2 \setminus Q)$, i.e., a large rectangle, of which the area, which did not belong to Q is cut out (light blue channel in Figure 2a). If Q is the only polygon of its color $f(Q)$ we call this setting Simply Bounded Minimum Link Fencing (SMLF). Moreover, if in an instance of SMLF we have $CH(\bigcup_{i=1}^{\kappa} \mathcal{P}_i \setminus Q) \subset \mathbb{R}^2 \setminus Q$, i.e., the convex hull of all input polygons except Q does not intersect Q, we speak of Convex Bounded Minimum Link Fencing (CMLF). The differences are illustrated in Figure 2.

1.2 Related Work

Despite the fact that the problem is natural and fundamental, little previous work exists. The problem of *enclosing* a set of objects by a shortest system of fences has recently been considered with a single set B_1 [1]. The task is to "enclose" the components of B_1 by a shortest system of fences. This can be formulated as a special case of our problem with $\kappa = 2$ colors: We add an additional set B_2, far away from B_1 and large enough so that it is never optimal to surround B_2. Thus, we have to enclose all components of B_1 and separate them from the unbounded region. In this setting, there will be no nested fences. Abrahamsen et al. [1] gave an $O(n \text{ polylog } n)$-time algorithm for inputs that consist of n unit disks.

Some variations with additional constraints on the fence become NP-hard already for point objects with two colors. For example, if we require the fence to be a single closed curve, it has been observed by Eades and Rappaport [6] already in 1993 that one can model the Euclidean Traveling Salesman Problem of computing the shortest tour through a given set of sites by placing two tiny objects of opposite color next to each site. If we require the fence to be connected, the same construction will lead to the Euclidean Steiner Tree Problem, which was shown to be NP-hard by Garey et al. in 1977 [8].

Polygon Nesting & Separation. Polygon nesting is considered to be a fundamental problem in computational geometry, and has been extensively studied since its inception. Aggarwal et al. [3] considered the problem of finding a polygon nested between two given convex polygons that has a minimal number of vertices. They gave an $O(n \log k)$ time algorithm for solving the problem, where n is the total number of vertices of the given polygons, and k is the number of vertices of a minimal nested polygon. Das [5] considered a variant of MLF in his thesis, which restricts every fence to enclose exactly one polygon, and showed that the problem is NP-hard. Given a polygon Q of m vertices inside another polygon P of n vertices, Ghosh [9] gave an $O((n+m) \log k)$ time algorithm for constructing a minimum nested convex polygon, where k is the number of vertices of the output polygon, improving upon the $O((n+m) \log(n+m))$ time algorithm of Wang and Chan [14]. However, on the other hand, given a family of disjoint polygons $P_1, P_2, \ldots, P_k$ in the plane, and an integer parameter m, it is NP-complete to decide if the P_i's can be pairwise separated by a polygonal family with at most m edges. Mitchell and Suri [12] presented efficient approximation algorithms for constructing separating families of near-optimal size.

Full proofs of statements marked by ($\star$) are found in the full paper [4].

2 Two-colored BMLF is NP-hard

In this section we will call polygons of color 1 *boundary polygons* and polygons of color 2 *inner polygons*. An instance of planar $3,4$-SATconsists of a Boolean CNF-formula ϕ with a set of variables $\mathcal{V} = \{v_1, \ldots, v_n\}$ and a set of clauses $\mathcal{C} \subset 2^{\mathcal{V}}$, s.t. every clause is a disjunction of three literals and every variable occurs at most four times as a literal in a clause. Additionally, we are given the embedded plane incidence graph $G_\phi = (\mathcal{V} \cup \mathcal{C}, E)$, where $E = \{vc \mid v \in \mathcal{V}, c \in \mathcal{C}, v \text{ occurs as a literal in } c\}$. It is known that deciding if a $3,4$-SAT-formula has a satisfying assignment is NP-complete [11].

Given an instance of planar $3,4$-SAT we create an instance of 2-colored BMLF $\mathcal{P}$, emulating the shape of G_ϕ with one unbounded outer polygon Q and multiple boundary polygons of the same color $f(Q) = 1$ (Figure 3), s.t. ϕ is satisfiable if and only if there exists a minimum link fencing for $\mathcal{P}$ with at most a certain fixed number of total segments.

Note that each gadget is described as a basic construction of gray polygons, in which inner polygons are placed. This is possible, because we will invert all gray polygons at the end of the reduction, s.t. the area of their union makes up exactly the actual free space of our entire construction, see Figure 3. Stating that fences are computed inside the gray polygons should be understood as fences being placed in the free space between polygons. Throughout this reduction we distinguish fences based on the inner polygons they include. We call two fences F and F' *congruent*, if and only if they enclose the same set of inner polygons. We call two fencings $\mathcal{F}$ and $\mathcal{F}'$ *congruent* if there is a bijective mapping $f : \mathcal{F} \to \mathcal{F}'$, s.t., every $F \in \mathcal{F}$ is congruent to $f(F) \in \mathcal{F}'$.

Let $\mathcal{P}$ be an instance of BMLF and S_1, S_2, and S_3 disjoint connected subsets of $\mathbb{R}^2 \setminus \cup_{P \in \mathcal{P}} P$. We call the ordered set $\mathcal{S} = \{S_1, S_2, S_3\}$ a *non-collinear triple* if there are no three points $p_1 \in S_1$, $p_2 \in S_2$, and $p_3 \in S_3$ such that the straight-line segment s from p_1 to p_3 contains p_2 and s lies completely inside $\mathbb{R}^2 \setminus \cup_{P \in \mathcal{P}} P$. The choice of S_2 only matters if there exists a straight-line segment in $\mathbb{R}^2 \setminus \cup_{P \in \mathcal{P}} P$ connecting points in S_1 and S_3. Therefore we can often omit S_2 from the description of the triple or assume it as arbitrarily chosen. We call S_2 the *bend-set* of $\mathcal{S}$. Let $\mathcal{F}$ be a fencing of $\mathcal{P}$, and S_1, S_2, and S_3 a non-collinear triple. We say a fence $F \in \mathcal{F}$ *crosses* the triple $\{S_1, S_2, S_3\}$ if the boundary of F contains at least one point p_i from each set S_i for $i = 1, 2, 3$ and there is a cyclic traversal of the boundary of F in which we see first p_1, then p_2 and finally p_3. We write $]S_1, S_3[$ to denote the part of the boundary of F that lies in between p_1 and p_3 and contains p_2.

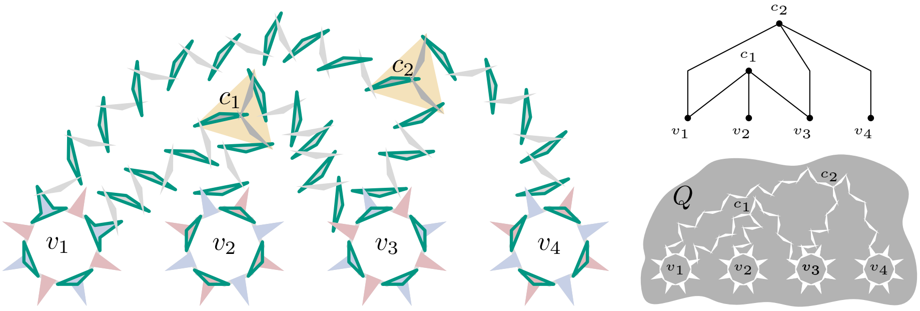


Figure 3 A (schematized) complete construction for a small instance $(v_1 \vee v_2 \vee \neg v_3) \wedge (v_1 \vee v_3 \vee v_4)$. The incidence graph is shown in the top right. Fences are highlighted in green. Also note that the boundary polygons make up most of the available area including an unbounded outer polygon Q as shown in the bottom right corner. For better readability, we will invert these colors in all subsequent figures.

▶ **Observation 2.** *Any fence in a fencing for an instance of BMLF crossing a non-collinear triple $\{S_1, S_2, S_3\}$ contains at least one bend in the interval $]S_1, S_3[$.*

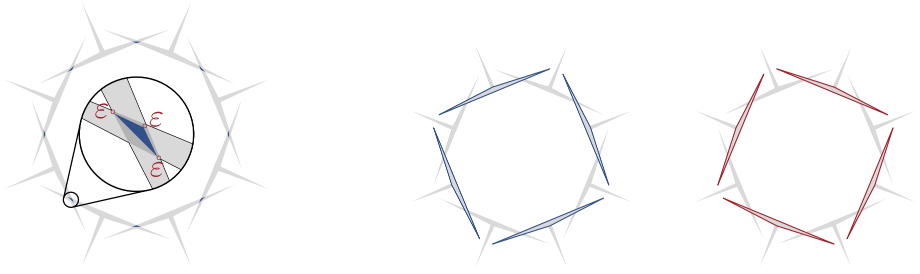
For $t > 0$ let $\mathcal{S}_1, \ldots, \mathcal{S}_t$ be non-collinear triples that are crossed by a fence F of a fencing $\mathcal{F}$ for some instance of BMLF. Let $\mathcal{S}_i = \{S_j, S_{j+1}, S_{j+2}\}$ for $i = 1, \ldots, t$ and $j = 3(i-1)+1$. We say that the triples are crossed by F *in-order* if there exist points p_i on F such that p_i is in the bend-set of $\mathcal{S}_i$ and there exists a cyclic traversal of F in which we see the points p_i in order of their indices. Without loss of generality we will assume throughout that when F crosses $\mathcal{S}_1, \ldots, \mathcal{S}_t$ in-order it always crosses for some $\mathcal{S}_i$ first the set S_i, then the bend-set S_{i+1}, and finally S_{i+2}. We write $]S_a, S_b[$ with $a < b$ and $a = 1, \ldots, 3t-1$ for the part of the boundary of F that lies between a point $p_a \in S_a$ and $p_b \in S_b$ such that there exist points $p_a, \ldots, p_b$ with $p_i \in S_i$ that we see in this order in a cyclic traversal of F. For a segment s of F we say it is *completely* contained in $]S_a, S_b[$ if the start- and endpoint of s are contained in $]S_a, S_b[$ for any choice of points p_a and p_b.

We say two non-collinear triples $\mathcal{S} = \{S_1, S_2, S_3\}$ and $\mathcal{S}' = \{S'_1, S'_2, S'_3\}$ are *non-overlapping* if there exist no two segments that intersect all six elements of $\mathcal{S} \cup \mathcal{S}'$ in order S_1, S_2, S_3, S'_1, S'_2, S'_3. In other words we require at least three different straight-line segments to connect a point $p_1 \in S_1$ with a point $p_6 \in S'_3$ and containing points $p_2 \in S_2$, $p_3 \in S_3$, $p_4 \in S'_1$, and $p_5 \in S'_2$ in order of their indices. Observe that by this definition the non-collinear triples $\{S_1, S_2, S_3\}$ and its reverse $\{S_3, S_2, S_1\}$ are non-overlapping. For a sequence of non-collinear triples $\mathcal{S}_1, \ldots, \mathcal{S}_t$ we say that the triples are non-overlapping if $\mathcal{S}_i$ is non-overlapping with $\mathcal{S}_{i+1}$ for $i = 1, \ldots, t+1 \mod t$.

Observation 2 together with the definition of non-overlapping gives the following.

▶ **Observation 3.** *Any fence in a fencing for an instance of BMLF crossing $t > 0$ non-overlapping non-collinear triples $\mathcal{S}_i = \{S_j, S_{j+1}, S_{j+2}\}$ for $i = 1, \ldots, t$ and $j = 3(i-1)+1$ in-order contains at least t bends and therefore at least $t-1$ complete straight-line segments. in the interval $]S_1, S_{3t}[$.*

We can now show a lower bound for the number of links a minimum-link fence uses in any solution of a BMLF instance. The lower bound essentially follows from Observation 3 after observing that the segment closing the fence can never reuse one of the $t-1$ segments that lie completely inside the sequence of non-collinear triples.



(a) Variable gadget construction with ε-gaps. **(b)** Fencing for *true*-state. **(c)** Fencing for *false*-state.

Figure 4 The variable gadget and its two possible fencings.

▶ **Lemma 4** ($\star$). *Let $\mathcal{P}$ be an instance of BMLF and $\mathcal{F}$ a minimum-link fencing for $\mathcal{P}$, then any fence $F \in \mathcal{F}$ that crosses $t > 0$ non-overlapping non-collinear triples in-order consists of at least t straight-line segments.*

2.1 Variable gadget

Every variable gadget consists of eight *T-polygons* (two per clause in which the variable can occur). Figure 4a illustrates the construction; T-polygons are marked in gray. Every T-polygon has an isosceles triangle as the *arm* of the T (the horizontal part of the T shape) and a *spike* (alternatively called a *true spike* and a *false spike*) protruding from the arm and two consecutive polygons overlap at the end of their arms. For every variable $v \in \mathcal{V}$, we construct a variable gadget $\mathcal{G}(v)$ as a circular arrangement of eight overlapping T-polygons.

For every pair of overlapping T-polygons A and B, we place an inner polygon P, s.t. $P \subset A \cap B$. Let us fix some A, B, and P as above, then we place P such that its three corner points have only a very small distance $\varepsilon > 0$ to some corner point of $A \cap B$. All three ε-length segments between a corner point of P and the closest corner point of $A \cap B$ have to be crossed by every fence enclosing P.

Crucially, the variable gadget has only two minimum link fencings. These two states are shown in Figure 4. We associate the one shown in Figure 4b to the variable gadget encoding the value *true* and the one shown in Figure 4c to encoding *false*.

▶ **Lemma 5** ($\star$). *There are exactly two minimum link fencings $\mathcal{F}_t$ and $\mathcal{F}_f$ of the variable gadget, both of which will enclose only triangles in the same T-polygon with each fence, resulting in a fencing with* 12 *links for the whole variable gadget, s.t. every other minimum link fencing is congruent to either $\mathcal{F}_t$ or $\mathcal{F}_f$.*

2.2 Clause gadget

For every clause $c \in \mathcal{C}$ in which three variables v_1, v_2, v_3 occur either as a positive or a negative literal, we create a clause gadget $\mathcal{G}(c)$. A clause gadget consists of three chains of an even number of gray triangles. These triangles are placed s.t. their hypotenuses intersect at an angle of at most π as shown in Figure 5a. The triangles are sufficiently long and thin, s.t., we can define two sets in every gray triangle (one to either side of the central line), s.t., the second set a' of the i-th triangle and the first set b of the $(i+1)$-th triangle form a non-collinear triple. By construction the non-collinear triple between the $(i-1)$-th and the i-th triangle and the one between the i-th and the $(i+1)$-th triangle are non-overlapping.

We place the three chains such that the three first triangles of the chains have a common intersection. Moreover, they intersect in such a way that their hypotenuses pairwise form $\frac{2\pi}{3}$ angles (Figure 5b). The last gray triangle of the first, second and third chain intersect a spike

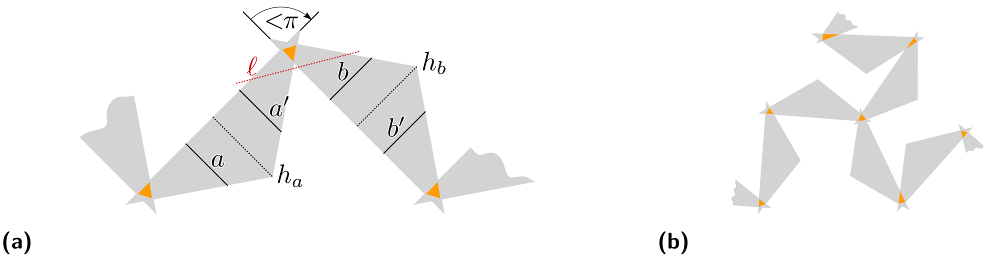


Figure 5 Wires are constructed from consecutive gray triangles places such that two consecutive triangles always contain a non-collinear triple, which are pairwise non-overlapping.

of $\mathcal{G}(v_1)$, $\mathcal{G}(v_2)$ and $\mathcal{G}(v_3)$, respectively. They intersect a true or false spike if the variable occurs as a positive or negative literal, respectively. We refer to each chain of gray polygons as a *wire*. The *length* of a wire is the number of gray triangles in its corresponding chain.

Let W_1, W_2, and W_3 be the wires of a clause gadget $\mathcal{G}(c)$ for clause c, where W_i intersects the spike of $\mathcal{G}(v_i)$ for $i \in \{1, 2, 3\}$. We place an inner triangle, denoted the *clause triangle* B_c of $\mathcal{G}(c)$, in the overlap of W_1, W_2, and W_3. Moreover, for wire W_i with gray triangles T^i_j we place inner triangles B^i_j in the overlap of the j-th and $(j+1)$-th gray triangle of the respective wire and a final triangle in the intersection with the spike of $\mathcal{G}(v_i)$. In the following we write $T_1, \ldots, T_k$ for the gray triangles and $B_1, \ldots, B_k$ for the inner polygons of one wire W_i, if i is clear from the context. Hence, inner triangle B_i is contained in the gray triangles T_i and T_{i+1} and gray triangle T_i for $i > 1$ contains the inner triangles B_i and B_{i+1}.

Let $B_1, \ldots, B_k$ be the inner polygons of a wire and F a fence containing B_i and B_j for some $i < j - 1$ and $i = 1, \ldots, k - 2$ but not B_z for $i < z < j$, then we say F *bypasses* B_z. For indices $1 \leq i_1 < i_2 < j_1 < j_2 \leq k$, we say two fences F_1 and F_2 containing some polygons of the wire *interleave* if B_{i_1} and B_{j_1} are in F_1 and F_1 bypasses B_{i_2} as well as B_{i_2} and B_{j_2} are in F_2 and F_1 bypasses B_{j_1}.

Let F be a fence of a minimum link fencing $\mathcal{F}$ for a clause gadget $\mathcal{G}(c)$. Let s be a segment contained in the union of the gray triangles that form $\mathcal{G}(c)$ such that F crosses s in two points p and q. Then *splitting* F at s means the following. Delete F in an ε-region around p and q this creates two polygonal-chains, say F' and F'' with endpoints p' and q' on one side of s and p'' and q'' on the other. Connect p' with q' and p'' with q'' to form the two new fences F' and F''. Clearly, $|F'| + |F''| = |F| + 2$.

One isolated wire

For the following we fix an arbitrary clause c. Let $\mathcal{G}(c)$ be the clause gadget of c and W one of the wires of $\mathcal{G}(c)$ with inner polygons $B_1, \ldots, B_k$. We denote as *isolated wire* the gray triangles of the chain of W that do not contain the clause triangle.

We are interested in how a minimum link fencing of an isolated wire looks like. Crucially, we first show that a fence of a minimum link fencing of an isolated wire cannot bypass any inner polygon of an inner polygon.

▶ **Lemma 6** ($\star$). *A minimum link fence $\mathcal{F}$ of an isolated wire W of $\mathcal{G}(c)$ does not contain a fence $F \in \mathcal{F}$ such that F bypasses an inner polygon B_i with $i \in \{2, \ldots, k-1\}$ of W.*

In the following we are going to bound the number of consecutive polygons that are contained in one minimum link fence of an isolated wire. We compare this then to a fence containing all inner polygons of an isolated wire. Such a fence, by construction, contains

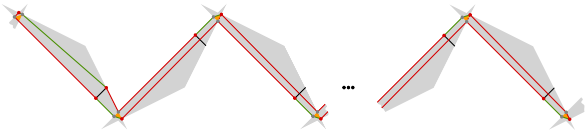


Figure 6 Non-colinear triples in a fence including a series of consecutive inner polygons.

$2z$ non-collinear triples and hence requires $2z$ segments by Lemma 4. Figure 6 shows these triples. Constructing such a fence is straight-forward by following these non-collinear triples. The following lemma summarizes this statement.

▶ **Lemma 7.** *Let $\mathcal{F}$ be a minimum link fencing of an isolated wire W of $\mathcal{G}(c)$, any fence $F \in \mathcal{F}$ that contains $z > 2$ consecutive inner polygons of W has at least $2z$ segments and such a fence exists.*

▶ **Lemma 8.** *Let $\mathcal{F}$ be a minimum link fencing of an isolated wire W of $\mathcal{G}(c)$, then every fence of $\mathcal{F}$ contains at most three consecutive inner polygons.*

Proof. Let $B_1, \ldots, B_k$ be the inner polygons of W. By Lemma 6 we can assume that the inner polygons of W contained in F are consecutive in the sequence of inner polygons. Let $F \in \mathcal{F}$ be a fence containing $z > 3$ inner polygons of W.

By Lemma 7 we know that F consists of $2z$ segments. We replace F with a fence F_1 including the two first polygons included in F and a fence F_2 including all $z - 2$ following inner polygons. Again by Lemma 7 it follows that $|F_2| = 2z - 4$ and it holds that $|F_1| = 3$. In sum, we get that $|F_1| + |F_2| = 2z - 4 + 3 = 2z - 1 \leq |F|$, a contradiction. ◀

Lemmas 7 and 8 now lead to a characterization of minimum link fences of isolated wires.

▶ **Lemma 9** (⋆). *Let $\mathcal{F}$ be a minimum link fencing of an isolated wire W of $\mathcal{G}(c)$ with k inner polygons, then $\mathcal{F}$ has in total $3k/2$ segments and $F \in \mathcal{F}$ contains exactly two consecutive inner polygons B_i and B_{i+1} for i odd.*

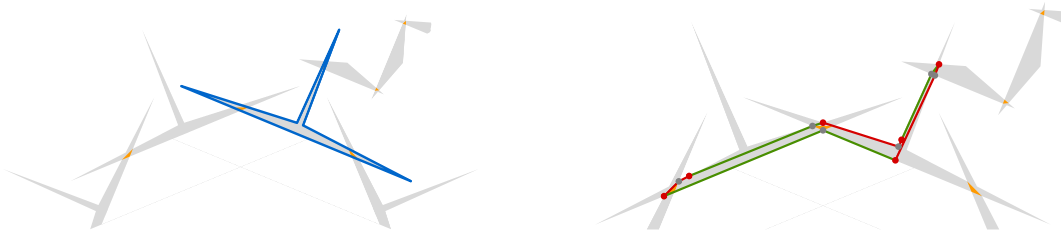
Integrating the clause triangle

So far we only considered one arbitrary isolated wire of $\mathcal{G}(c)$. To put things together we need to consider the interaction of the three wires of $\mathcal{G}(c)$. Specifically, we need to show that no fence in a minimum link fencing of $\mathcal{G}(c)$ contains inner polygons from two different wires.

We extend the definition of bypassing an inner polygon of a wire to a whole clause gadget. Let F be a fence for $\mathcal{G}(c)$, then F *bypasses* an inner polygon B^i_j of wire W_i of $\mathcal{G}(c)$ if F contains the clause triangle B_c or some inner polygon of a wire $W_{i'}$ with $i' \neq i$ and F contains B^i_l for wire W_i with $l > j$. We say F *bypasses* the clause triangle of $\mathcal{G}(c)$ if F contains inner polygons of at least two different wires of $\mathcal{G}(c)$ but not the clause triangle B_c of $\mathcal{G}(c)$.

As for an isolated wire we can show that no inner polygon for a whole clause gadget can be bypassed. This can be seen after observing that no fence can bypass the inner polygons of an isolated wire without violating Lemma 6. The remainder of the proof is then a careful case enumeration, which can be found in the full paper [4].

▶ **Lemma 10** (⋆). *Let $\mathcal{F}$ be a minimum link fencing of $\mathcal{G}(c)$ and $B_1, \ldots, B_k$ the inner polygons of one of the wires of $\mathcal{G}(c)$. Then there is no fence $F \in \mathcal{F}$ that bypasses an inner polygon B_i with $i \in \{1, \ldots, k-1\}$.*



(a) $\mathcal{G}(v)$ in correct state.

(b) $\mathcal{G}(v)$ in incorrect state.

Figure 7 If $\mathcal{G}(v)$ is in the correct truth state (a) inclusion of the first inner polygon of $\mathcal{G}(v, c)$ induces an additional cost of two links, otherwise (b) the additional cost is at least three.

Finally, we show that no minimum link fence of a clause gadget can ever fence two polygons that are in different wires. Again, this is shown essentially via a case enumeration that considers how a minimum link fence includes the first two to three polygons of each wire together with the clause triangle. In each case we can conclude that there exists a fence with fewer segments that in fact does not use the inner polygons of two distinct wires.

▶ **Lemma 11** ($\star$). *Let $\mathcal{F}$ be an optimal fencing of a clause gadget, then there exists no fence $F \in \mathcal{F}$, which includes inner polygons belonging to two different wires.*

We can now use Lemma 11 to argue that the clause triangle is only included in a fence together with inner polygons of at most one wire. We say that such a wire is in a *satisfying* state. The other two wires should therefore, by Lemma 9, only use fences including two inner polygons; leading to $\frac{3(k_a+k_b+k_c)}{2} + 3$ segments in total (k_a, k_b and k_c being the number of inner polygons in the wires). If we include the clause triangle in a fence of a wire, we get the same amount of segments, however, we can choose fences, s.t., the last inner polygon of the wire which fences the clause triangle, is fenced alone. This will be crucial in the argument of how the wires and therefore the clause gadget interacts with the variable gadget.

Interaction with the variable gadgets

It remains to describe the interaction between the variable and clause gadgets. Depending on the state of the variable gadget we can fence the last inner polygon of a wire in the fence of a variable gadget. We provide a fence with 5 segments (i.e., two additional ones) for the case, where the variable gadget is in the correct state and the existence of six non-collinear triples for the other case, see Figure 7.

▶ **Lemma 12.** *The last inner polygon of a wire can be included in a fence of the variable gadget, whose spike it is connected to for the cost of two additional segments if the variable gadget is in the correct state and at least three additional segments otherwise.*

Concluding the interaction between clause and variable gadget we show that given a variable gadget is in the correct state w.r.t. a clause gadget we can fence the inner polygons of the wires of a clause gadget using 3/2 segments per polygon and adding only two segments to the fence of the variable gadget.

▶ **Lemma 13** ($\star$). *If and only if at least one of the connected variable gadgets is in the correct state, the clause gadget can be fenced with a total of $\frac{3(k_a+k_b+k_c)}{2}$ segments plus two additional segments to a fence of the variable gadget, which is connected to the wire in the satisfying state.*

Correctness

It remains to argue the correctness of our reduction which then implies our main theorem.

▶ **Theorem 14** ($\star$). *Two-colored BMLF is NP-hard even when restricting all fences to include at most three polygons.*

Proof sketch. For an instance ϕ of planar 3, 4-SAT, we construct a variable gadget for every variable and connect the clause gadgets accordingly. By construction any fencing with $|\mathcal{V}| \cdot 12 + \sum_{c \in \mathcal{C}}(\frac{3(k^c)}{2} + 2)$ segments, requires one wire of every clause gadget to be in a satisfying state. The connected variable gadget is forced into the true or false state, depending on the connected spike. This implies a satisfying variable assignment for ϕ.

Conversely, since every variable is either true or false and for every clause there is a true literal, we can set all variable gadgets into the true or false state according to the assignment and are guaranteed to be able to put exactly one wire per clause gadget into a satisfying state for an additional cost of exactly two. ◀

3 An XP-algorithm for BMLF with at most two polygons in each fence

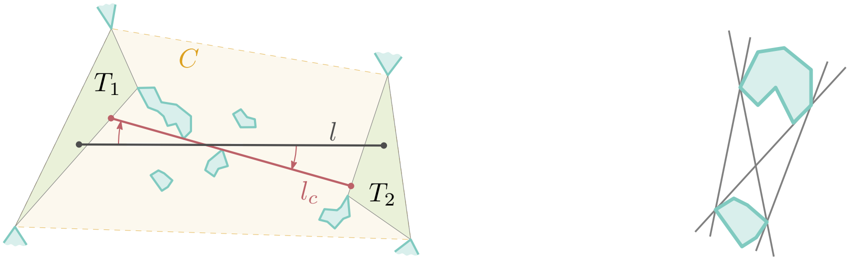
In Section 2 we showed that BMLF is NP-hard when there are only two colors, each fence contains at most three polygons, and each fence consists of at most five links. In contrast, we are going to show in this section that BMLF can be solved in XP-time when parameterizing the problem by the maximum number of links in any fence and allowing at most two polygons per fence, i.e., the problem can be solved in polynomial-time when fixing the maximum number of links in any fence and restricting each fence to contain at most two polygons.

For our algorithm we make use of the following result derived from the work of Hershberger and Snoeyink [10]. It allows us to compute for a given *loop*, i.e., a closed polygonal curve, inside a polygon with holes, a minimum-link loop of the same homotopy in time $O(nk)$, where n is the complexity of the polygon and k is the size of the resulting fence.

▶ **Theorem 15** (Derived from Section 5.2 [10]). *Given a polygon P without self-intersections but potentially with holes of complexity n, an integer k, and a loop α lying in the interior of P with $O(nk)$ corners, we can decide in time $O(nk)$ if there exists a loop α' of the same homotopy-class as α with at most k links.*

▶ **Remark 16.** *It is worth noting that in the paper by Hershberger and Snoeyink [10] Theorem 15 is only stated in text. The runtime is given as $O(C_\alpha + \Delta_\alpha + \Delta_{\alpha'})$, where C_α is the complexity of α, the free space between polygons is assumed to be triangulated and Δ_α and $\Delta_{\alpha'}$ are the number of triangulation edges intersected by α and the fence α', respectively. However an example of an instance with multiple obstacles is given, in which $\Delta_{\alpha'} \in \Omega(nk)$, where n is the number of corners over all polygons. Since in our scenario we can find a path α s.t. $C_\alpha \in O(nk)$ and $\Delta_\alpha \in O(nk)$, we can make the assumption that α''s complexity is in $O(nk)$.*

Let P be a polygon without self-intersections. We denote with $\mathcal{T}_P = \{T_1, \dots, T_z\}$ a triangulation of P with triangles $T_1, \dots, T_z$. Note that we do not require any further properties of $\mathcal{T}_P$. If P is clear from the context we omit it and set $\mathcal{T} = \mathcal{T}_P$. Let $T_1, T_2 \in \mathcal{T}$ be two triangles and let l be a line segment with endpoints p and q such that $p \in T_1$ and $q \in T_2$. We call l a *splitting segment*. Consider Figure 8a for an example for T_1 and T_2 if l contains no points of $\mathbb{R}^2 \setminus P$. Intuitively, a splitting segment separates the holes that intersect the convex hull of $T_1 \cup T_2$ into two sets. Let $\mathcal{H}$ be all the holes of P that intersect or are fully



(a) A splitting segment l_c equivalent to l and bitangent to two holes. **(b)** Bitangents.

Figure 8 **(a)** We can obtain the splitting segment l_c from a splitting segment l by rotating l clockwise until it is a bitangent to two holes. **(b)** Any pair of two polygons admits at most four bitangents, only one of which can not be rotated clockwise without intersecting one of the polygons.

contained in the interior of the convex hull of $T_1 \cup T_2$. We say that a hole $H \in \mathcal{H}$ is to the *left* (*right*) of l if the from p to q oriented supporting line of l leaves H in the left (right) half-plane. We call two splitting segments of T_1 and T_2 *equivalent* if the same holes of $\mathcal{H}$ are to their respective left and right. Segments which intersect holes are not splitting segments.

▶ **Lemma 17** ($\star$). *Let P be a polygon without self-intersection, $\mathcal{H}$ a set of holes and $\mathcal{T}$ a triangulation of P. Then for every pair of triangles $T_1, T_2 \in \mathcal{T}$ with $T_1 \neq T_2$ there are at most $4|\mathcal{H}|^2$ different equivalence classes of splitting segments.*

▶ **Theorem 18** ($\star$). *Given an instance $\mathcal{P}$ of BMLF with outer polygon $Q \in \mathcal{P}$, we can decide in time $O(kn^{2k+4})$ if a minimum link fencing $\mathcal{F}$ of $\mathcal{P}$ exists, in which every fence contains at most two polygons, each fence in the fencing has at most k segments, and n is the number of corners in $\mathcal{P}$.*

Proof sketch. For each polygon and for each pair of polygons we compute a minimum-link fence. Let λ_{uv} be the number of links for a minimum link fence containing $P_u, P_v \in \mathcal{P}$ and λ_u the number of links for a minimum link fence containing only $P_u \in \mathcal{P}$. Consider a complete graph G containing one vertex u for each polygon $P_u \in \mathcal{P}$ and one more vertex x if $|\mathcal{P}|$ is odd. Set the edge-weights $w(u, v) = \min\{\lambda_{uv}, \lambda_u + \lambda_v\}$ and $w(x, u) = \lambda_u$ for $P_u, P_v \in \mathcal{P}$. If for some $P_u \in \mathcal{P}$ or pair $P_u, P_v \in \mathcal{P}$ no fence with $\leq k$ segments exists we remove that edge. Computing a minimum weight perfect matching in this graph yields a minimum link fencing.

It remains to compute the minimum-link fences for each polygon and for each pair of polygons of $\mathcal{P}$. We consider a triangulation $\mathcal{T}$ of the free space of $\mathcal{P}$. For one single polygon we construct a plane loop around it by just traversing the incident triangles in the triangulation. To compute the minimum link fence for a pair of polygons in $\mathcal{P}$ we need to do more work. Since $\mathcal{T}$ contains only $O(n)$ triangles we can branch over the $O(n^k)$ ordered k-tuples of triangles. Moreover, by Lemma 17 we can branch over the $O(n^{2k})$ different splitting segments. If for our choice of triangles all splitting segments between consecutive triangles exist we construct a plane loop α if possible or otherwise reject the branch.

Let $T_1, \ldots, T_k$ be the chosen k-tuple of triangles and $l_1, \ldots, l_k$ the splitting segments. If none of the splitting segments intersect the same triangles in between two consecutive triangles T_i and T_{i+1} this is straight forward. If there are triangles that are intersected multiple times we have to evaluate $2^{O(k)}$ choices of how to resolve the self-crossings such a repetition induces for the loop α. For each valid choice we apply Theorem 15.

These are only $O(kn^{2k+4})$ choices in total and computing a minimum weight perfect matching can be done in $O(V^2E)$ time (with V being the number of vertices and E the

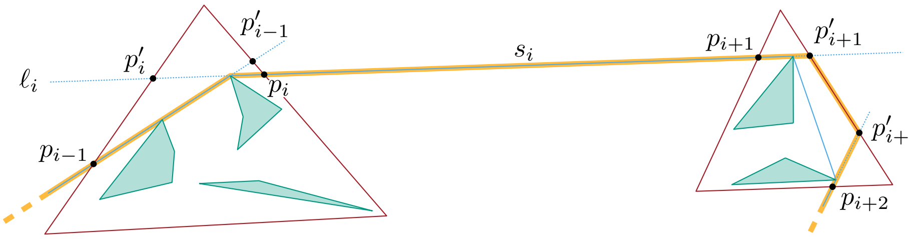


Figure 9 Computing a new fence (orange) from the old fences (purple) and the convex hull (blue).

number of edges) via finding a maximum weight perfect matching (e.g. [7]) on the same graph with edge weights set to maximum edge-weight plus one minus the original edge-weight. ◀

4 An algorithm for two-colored CMLF

In this section we present an algorithm for solving two-colored CMLF. Computing a minimum-link fence in this setting can be done by computing a fence for the convex hull of the contained polygons with the algorithm by Wang [13] which runs in time $O(n \log n)$ with n being the number of corners of the contained polygons. Throughout this section an instance of CMLF is given as $(\mathcal{P}, Q)$ where Q is the outer polygon and $\mathcal{P}$ is the set of polygons contained in Q.

▶ **Lemma 19.** *Given an instance $(\mathcal{P}, Q)$ of two-colored CMLF, let $\mathcal{F}$ be a solution for $(\mathcal{P}, Q)$. There exists a solution $\mathcal{F}'$ for the two-colored CMLF instance $(CH(\mathcal{P}), Q)$ with $|F| = |F'|$.*

Proof. As F is a minimum-link fencing of $(\mathcal{P}, Q)$, it suffices to consider the case where a minimal link fencing of $(CH(\mathcal{P}), Q)$ has strictly more segments than $|F|$. We will construct a new fence F° from this instance. Let $(p_1, \ldots, p_z)$ be the intersection points between F and $CH(\mathcal{P})$ ordered as they appear in a clockwise traversal of the convex hull, and observe that z is even. Let p_i, p_{i+1} be pairs of intersection points between F and $CH(\mathcal{P})$ such that the straight-line segment s_i connecting p_i and p_{i+1} lies on $CH(\mathcal{P})$ and completely outside of F (see Figure 9). Consider the supporting line ℓ_i of s_i. If the fence lies completely in one of the closed half-planes bounded by ℓ_i we add s_i to F°. Assume this is not the case. As s_i is on $CH(\mathcal{P})$ we get that ℓ_i does not intersect any polygon in $\mathcal{P}$. Moreover, as $\mathcal{F}$ consists of closed simple polygons we find two intersection points p'_i and p'_{i+1} that lie on ℓ_i, s.t., the parts of F appearing in a clockwise traversal from p'_i to p_i, as well as the ones in a clockwise traversal from p_{i+1} to p'_{i+1} lie outside of $CH(\mathcal{P})$. We add the segment s'_i between p'_i and p'_{i+1} to F°. Doing this for every pair of intersections we obtain a set of segments F°, where all segments are on the convex-hull of $\mathcal{P}$. Note that it is possible for these segments to intersect; if that is the case we only keep the parts until their intersection point. Finally, the start and end-points of connected chains of segments in F° lie on segments of fences in $\mathcal{F}$. We can convert F° into a fence of $CH(\mathcal{P})$ by connecting these endpoints along the fences in $\mathcal{F}$ and that fence will be disjoint from $\mathcal{P}$ (except possibly touching $\mathcal{P}$ in corner points).

It remains to argue that indeed $|F^\circ| \leq |F|$. We partition F° into two categories, segments that coincide with segments in F and segments that do not. Each of them is either a full segment of F or originates from the intersection of at most two different s'_i's and a segment of F. Furthermore, we add $z/2$ segments s'_i that are not sub-segments of segments in F. For each such s'_i we find at least one segment of F for which we did not add any sub-segment to F°. These are the segments of F on which p_i and p_{i+1} lie or that are fully outside of F°. ◀

▶ **Theorem 20.** *Two-colored CMLF can be solved in time $O(n \log n)$ where n is the number of corners of polygons in $\mathcal{P}$.*

5 Conclusion

We have shown BMLF to be NP-hard even if every fence contains at most three polygons, each fence has at most five links, and only two different colors of polygons are present. Our reduction holds regardless of requiring disjoint fences or not. Note, that our reduction can be adapted to not require the outer bounding polygon Q. Instead, we can replace Q by one polygon with a narrow and very complex channel, connecting the "inside" with the "outside". On the algorithmic side, we gave an XP-algorithm for BMLF parameterized by the maximum number of links in a fence and allowing at most two polygons per fence. We also showed that two-colored CMLF can be solved in polynomial time.

It is open if one can eliminate the exponential dependency on the number of links in our algorithm for BMLF. Furthermore, while our reduction holds when replacing the outer bounding polygon, our algorithm does not since we cannot immediately apply Theorem 15. Similarly, requiring the fences to be disjoint for BMLF is an interesting open direction.

References

1 Mikkel Abrahamsen, Anna Adamaszek, Karl Bringmann, Vincent Cohen-Addad, Mehran Mehr, Eva Rotenberg, Alan Roytman, and Mikkel Thorup. Fast fencing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 564–573, 2018. doi:10.1145/3188745.3188878.

2 Mikkel Abrahamsen, Panos Giannopoulos, Maarten Löffler, and Günter Rote. Geometric multicut. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *LIPIcs*, pages 9:1–9:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.9.

3 Alok Aggarwal, Heather Booth, Joseph O'Rourke, Subhash Suri, and Chee-Keng Yap. Finding minimal convex nested polygons. *Information and Computation*, 83(1):98–110, 1989.

4 Sujoy Bhore, Fabian Klute, Maarten Löffler, Martin Nöllenburg, Soeren Terziadis, and Anaïs Villedieu. Minimum link fencing, 2022. arXiv:2209.14804.

5 Gautam Das. *Approximation schemes in computational geometry.* PhD thesis, University of Wisconsin, Madison, 1991.

6 Peter Eades and David Rappaport. The complexity of computing minimum separating polygons. *Pattern Recognition Letters*, 14(9):715–718, 1993. doi:10.1016/0167-8655(93)90140-9.

7 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.

8 Michael R Garey, Ronald L Graham, and David S Johnson. The complexity of computing steiner minimal trees. *SIAM journal on applied mathematics*, 32(4):835–859, 1977. doi:10.1137/0132072.

9 Subir Kumar Ghosh. Computing the visibility polygon from a convex set and related problems. *Journal of Algorithms*, 12(1):75–95, 1991. doi:10.1016/0196-6774(91)90024-S.

10 John Hershberger and Jack Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications*, 4:63–97, 1994. doi:10.1016/0925-7721(94)90010-8.

11 Klaus Jansen and Haiko Müller. The minimum broadcast time problem for several processor networks. *Theoretical Computer Science*, 147(1-2):69–85, 1995. doi:10.1016/0304-3975(94)00230-G.

12 Joseph SB Mitchell and Subhash Suri. Separation and approximation of polyhedral objects. *Computational Geometry: Theory and Applications*, 5(2):95–114, 1995. doi:10.1016/0925-7721(95)00006-U.

13 Cao An Wang. Finding minimal nested polygons. *BIT Computer Science section*, 31(2):230–236, 1991. doi:10.1007/BF01931283.

14 Cao An Wang and Edward P. F. Chan. Finding the minimum visible vertex distance between two non-intersecting simple polygons. In *Proceedings of the Second Annual ACM SIGACT/SIGGRAPH Symposium on Computational Geometry, Yorktown Heights, NY, USA, June 2-4, 1986*, pages 34–42. ACM, 1986. doi:10.1145/10515.10519.