**IMFM**
Institute of Mathematics, Physics and Mechanics
Jadranska 19, 1000 Ljubljana, Slovenia

**Preprint series**
**Vol. 49 (2011), 1152**
ISSN 2232-2094

# MINIMUM CELL CONNECTION AND SEPARATION IN LINE SEGMENT ARRANGEMENTS

Helmut Alt Sergio Cabello
Panos Giannopoulos Christian Knauer

Ljubljana, June 23, 2011

# Minimum cell connection and separation in line segment arrangements[*]

Helmut Alt[†][‡] Sergio Cabello[§][¶] Panos Giannopoulos[†][‡] Christian Knauer[‖][†]

**Abstract**

We study the complexity of the following cell connection and separation problems in segment arrangements. Given a set of straight-line segments in the plane and two points $a$ and $b$ in different cells of the induced arrangement:

(i) compute the minimum number of segments one needs to remove so that there is a path connecting $a$ to $b$ that does not intersect any of the remaining segments;
(ii) compute the minimum number of segments one needs to remove so that the arrangement induced by the remaining segments has a single cell;
(iii) compute the minimum number of segments one needs to retain so that any path connecting $a$ to $b$ intersects some of the retained segments.

We show that problems (i) and (ii) are NP-hard and discuss some special, tractable cases. Most notably, we provide a linear-time algorithm for a variant of problem (i) where the path connecting $a$ to $b$ must stay inside a given polygon $P$ with a constant number of holes, the segments are contained in $P$, and the endpoints of the segments are on the boundary of $P$. For problem (iii) we provide a cubic-time algorithm.

## 1 Introduction

In this paper we study the complexity of some natural optimization problems in segment arrangements. Let $S$ be a set of straight-line segments in $\mathbb{R}^2$, $\mathcal{A}(S)$ be the arrangement induced by $S$, and $a, b$ be two points not incident to any segment of $S$ and in different cells of $\mathcal{A}(S)$.

In the 2-Cells-Connection problem we want to compute a set of segments $S' \subseteq S$ of minimum cardinality with the property that $a$ and $b$ belong to the same cell of $\mathcal{A}(S \setminus S')$. In other words, we want to compute an $a$-$b$ path that *crosses* the minimum number of segments of $S$ counted without multiplicities. The *cost* of a path is the total number of segments it crosses.

[*]Part of the results contained here were presented at EuroCG 2011 [ACGK11]. Research partially conducted at the 9th McGill - INRIA Barbados Workshop on Computational Geometry, 2010.

[†]Institut für Informatik, Freie Universität Berlin, Takustraße 9, D-14195 Berlin, Germany, {alt, panos}@mi.fu-berlin.de

[‡]Research supported by the German Science Foundation (DFG) under grant Kn 591/3-1.

[§]Department of Mathematics, IMFM and FMF, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia, sergio.cabello@fmf.uni-lj.si.

[¶]Research was partially supported by the Slovenian Research Agency, program P1-0297.

[‖]Institut für Informatik, Universität Bayreuth Universitätsstraße 30 D-95447 Bayreuth Germany, christian.knauer@uni-bayreuth.de

In the All-Cells-Connection problem we want to compute a set $S' \subseteq S$ of minimum cardinality such that $\mathcal{A}(S \setminus S')$ consists of one cell only.

In the 2-Cells-Separation problem we want to compute a set $S' \subseteq S$ of minimum cardinality that *separates* $a$ and $b$, i.e., $a$ and $b$ belong to different cells of $\mathcal{A}(S')$ – equivalently – any $a$-$b$ path intersects some segment of $S'$.

Apart from being interesting in their own right, the problems we consider here are also natural abstractions of problems concerning sensor networks. Each segment is surveyed (covered) by a sensor, and the task is to find the minimum number of sensors of a given network over some domain that must be switched on or off so that: an intruder can be detected when walking between two given points (2-Cells-Separation), or can walk freely between two given points (2-Cells-Connection) or can reach freely any point (All-Cells-Connection). Because of these applications, it is worth considering a variant where the segments lie inside a given polygon $P$ with holes and have their endpoints on the boundary of $P$, and the $a$-$b$ path must also stay inside $P$. See Fig. 1 for an example of this last scenario. We refer to these variants as the restricted 2-Cells-Separation or 2-Cells-Connection in a polygon.

**Our results.** We provide an algorithm that solves 2-Cells-Separation in $\mathcal{O}(n^2+nk)$ time, where $k$ is the number of pairs of segments that intersect. The same algorithm, with an extra logarithmic factor, works for a generalization where the segments are weighted. The algorithm itself is simple, but its correctness is not obvious. We justify its correctness by considering an appropriate set of cycles in the intersection graph and showing that it satisfies the so-called *3-path condition* [Tho90] (see also [MT01, Chapter 4]). The use of the 3-path condition for solving 2-Cells-Separation is surprising and makes the connection to topology clear.

We show that both 2-Cells-Connection and All-Cells-Connection are NP-hard even when the segments are in general position. The first result is given by a careful reduction from Max-2-Sat, which also implies APX-hardness. The second one follows from a straightforward reduction that uses a connection to the feedback vertex set problem in the intersection graph of the segments and holds even if there are no proper segment crossings. Also, when any three segments may intersect only at a common endpoint, 2-Cells-Connection is fixed-parameter tractable with respect to the number of proper segment crossings.

Finally, we consider the restricted problems in a polygon. The restricted 2-Cells-Separation in a polygon is easily reduced to the general weighted version and thus can be solved efficiently. The restricted 2-Cells-Connection in a polygon remains NP-hard but can be solved in near-linear time for any fixed number of holes. The approach for this latter result uses homotopies to group the segments into clusters with the property that any cluster is either contained or disjoint from the optimal solution.

**Related work.** Our NP-hardness proof for 2-Cells-Connection has been carefully extended by Kirkpatrick and Tseng [Tse11], who showed that the 2-Cells-Connection remains NP-hard even for *unit-length* segments. However, their result does not imply APX-hardness for unit-length segments. The related problem of finding (from scratch) a set of segments with minimum total length that forms a barrier between two specified regions in a polygonal domain has been shown to be polynomial-time solvable by Kloder and Hutchinson [KH07].

The problems we consider can of course be considered for other geometric objects, most notably unit disks. To this end, closely related work was done by Bereg and Kirkpatrick [BK09], who studied the counterpart of 2-Cells-Connection in arrangements of unit disks and gave a 3-approximation algorithm. While the complexity of 2-Cells-Connection for unit (or

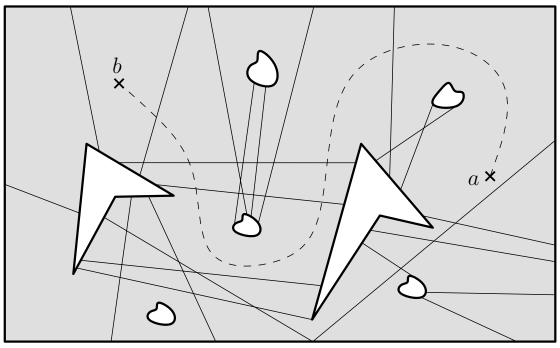


Figure 1: A polygon with holes and a minimum-cost $a$-$b$ path.

arbitrary) disks is still unknown, there exist polynomial-time algorithms for restricted belt-shaped and simple polygonal domains [KLA07]. Simultaneously and independently to our work, Gibson et al. [GKV11] have considered the problem of separating $k$ points in an arrangements of disks and provided a polynomial-time $O(1)$-approximation algorithm. Their approach is based on building a solution by considering several instances 2-Cells-Separation on arrangements of disks, which they can solve approximately.

## 2 Separating two cells

In this section we provide a polynomial-time algorithm for 2-Cells-Separation. We will actually solve a weighted version, where we have a weight function $w$ assigning weight $w(s) \geq 0$ to each segment $s \in S$. For any subset $S' \subseteq S$ we define its weight $w(S')$ as the sum of the weights over all segments $s \in S'$. The task is to find a minimum weight subset $S' \subseteq S$ that separates two given points $a$ and $b$. Our time bounds will be expressed as a function of $n$, the number of segments in $S$, and $k$, the number of pairs of segments in $S$ that intersect. We first describe the algorithm, and then justify its correctness.

We assume for simplicity of exposition that the segment $\overline{ab}$ is vertical and does not contain any endpoint of $S$ or any vertex of $\mathcal{A}(S)$.

Let $\gamma$ be a polygonal path contained in $\bigcup S$, possibly with self-intersections. Because of our assumption on general position, no vertex of $\gamma$ is on the segment $\overline{ab}$. We define $N(\gamma; a, b)$ as the number of oriented intersections of $\gamma$ with $\overline{ab}$: a crossing where $\gamma$ goes from the left to the right of $\overline{ab}$ contributes $+1$ to $N(\gamma; a, b)$, while a right-to-left crossing contributes $-1$ to $N(\gamma; a, b)$. We have $N(\text{reverse}(\gamma); a, b) = -N(\gamma; a, b)$.

In a graph, we will use the term *cycle* for a closed walk without repeated vertices. A polygonal path is *simple* if it does not have self-intersections.

### 2.1 The algorithm

From $S$ we construct its intersection graph $\mathbb{G} = (S, \{ss' \mid s \cap s' \neq \emptyset\})$. See Fig. 2(a)-(b). Note that $\mathbb{G}$ has $k$ edges. To each edge $ss'$ of $\mathbb{G}$ we attach a weight (abstract length) $w(s) + w(s')$. Any distance in $\mathbb{G}$ will refer to these edge weights. For any walk $\pi$ in $\mathbb{G}$ we use $\text{len}_{\mathbb{G}}(\pi)$ for its length, that is, the sum of the weights on its edges counted with multiplicity, and $S(\pi) = V(\pi)$ for the set of segments along $\pi$. For any spanning tree $T$ in $\mathbb{G}$ and any edge $e \in E(\mathbb{G}) \setminus E(T)$, let $\tau(T, e)$ denote the cycle obtained by concatenating the edge $e$ with

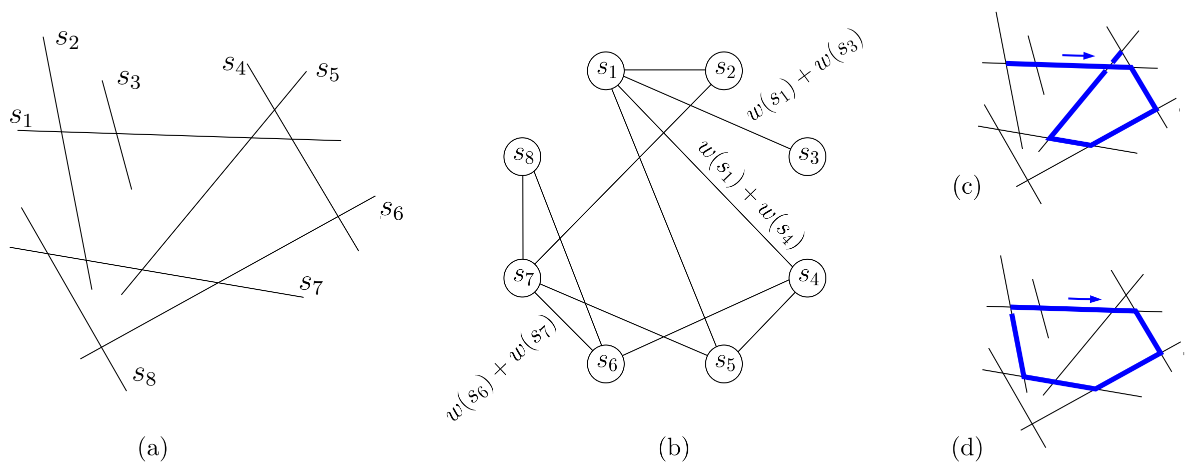


Figure 2: (a) A set of segments $S$. (b) The corresponding intersection graph $\mathbb{G}$ with some of its edge-weights. (c) The polygonal path $\gamma(\pi)$ for the walk $\pi = s_2s_1s_4s_6s_7s_5s_4$. (d) The closed polygonal path $\gamma(\pi)$ for the closed walk $\pi = s_2s_1s_4s_6s_7s_2$.

the path in $T$ connecting both endpoints of $e$. (The actual orientation of $\tau(T,e)$ will not be relevant.)

Consider any walk $\pi = s_0s_1\cdots s_t$ in $\mathbb{G}$. This walk defines a polygonal path, denoted by $\gamma(\pi)$, which has vertices $x_0, x_1, \dots, x_{t-1}$, where $x_j = s_j \cap s_{j+1}$ for $j = 0, \dots, t-1$. If $\pi$ is a closed walk with $s_0 = s_t$, then we take $\gamma(\pi)$ to be a closed polygonal path whose last edge is $x_{t-1}x_0$, which is contained in $s_0$. See Fig. 2(c)-(d). The polygonal path $\gamma(\pi)$ is contained in $\bigcup S(\pi)$. Note that even if $\pi$ is a cycle the closed polygonal path $\gamma(\pi)$ may have self-intersections.

For any segment $r \in S$, let $T_r$ be a shortest-path tree in $\mathbb{G}$ from $r$; if there are several we fix one of them. We will mainly use polygonal paths arising from cycles $\tau(T_r, e)$, where $e \in E(\mathbb{G}) \setminus E(T_r)$. Thus we introduce the notation $\gamma(r,e) = \gamma(\tau(T_r, e))$. (Again, the actual orientation of $\gamma(r,e)$ will not be relevant.)

The algorithm is the following. We compute the set

$$P = \{(r,e) \in S \times E(\mathbb{G}) \mid e \in E(\mathbb{G}) \setminus E(T_r) \text{ and } N(\gamma(r,e)); a, b) \neq 0\},$$

choose

$$(r^*, e^*) = \arg \min_{(r,e)\in P} \operatorname{len}_{\mathbb{G}}(\tau(T_r, e)),$$

and return $S(\tau(T_{r^*}, e^*))$. This finishes the description of the algorithm. For analyzing it, it will be convenient to use the notation $\tau^* = \tau(T_{r^*}, e^*)$ and $\gamma^*$ for the polygonal path $\gamma(\tau^*)$.

The algorithm, as described above, can be implemented in $\mathcal{O}(n^3k)$ time in a straightforward way. We can speed up the procedure to obtain the following result.

**Lemma 1.** *The pair $(r^*, e^*)$ can be computed in $\mathcal{O}(nk + n^2 \log n)$ time.*

*Proof.* The graph $\mathbb{G}$ can be constructed explicitly in $\mathcal{O}(n^2)$ time by checking each pair of segments, whether they cross or not. Recall that $\mathbb{G}$ has $k$ edges.

For any segment $r \in S$, let us define

$$\begin{aligned} E_r &= \{e \in E(\mathbb{G}) \setminus E(T_r) \mid (r,e) \in P\} \\ &= \{e \in E(\mathbb{G}) \mid e \in E(\mathbb{G}) \setminus E(T_r) \text{ and } N(\gamma(r,e)); a, b) \neq 0\}. \end{aligned}$$

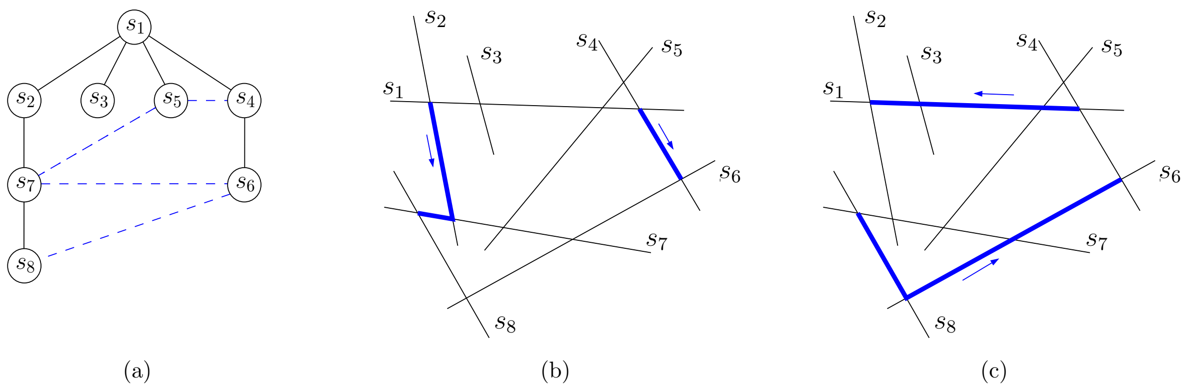


Figure 3: (a) Tree $T_{s_1}$ for the scenario of Fig. 2 assuming unit weights in the segments. In this case $C_{s_1}(s_8) = s_2$ and $C_{s_1}(s_6) = s_4$. (b) The polygonal paths $\gamma(T_{s_1}[s_8])$ and $\gamma(T_{s_1}[s_6])$. (c) The polygonal paths $\gamma(p_{s_1}(s_8)s_8s_6p_{s_1}(s_6)) = \gamma(s_7s_8s_6s_4)$ and $\gamma(C_{s_1}(s_6)s_1C_{s_1}(s_8)) = \gamma(s_4s_1s_2)$ that are used to compute $N(\gamma(s_1, s_6s_8); a, b)$ in Lemma 1.

Note that

$$P = \bigcup_{r\in S} \{r\} \times E_r,$$

and therefore

$$\min_{(r,e)\in P} \text{len}_{\mathbb{G}}(\tau(T_r, e)) = \min_{r\in S} \min_{e\in E_r} \text{len}_{\mathbb{G}}(\tau(T_r, e)).$$

Thus, $(r^*, e^*)$ can be computed by finding, for each $r \in S$, the value

$$\min_{e\in E_r} \text{len}_{\mathbb{G}}(\tau(T_r, e)).$$

We shall see that, for each fixed $r \in S$, such value can be computed in $\mathcal{O}(k + n \log n)$ time. It then follows that $(r^*, e^*)$ can be found in $|S| \times \mathcal{O}(k + n \log n) = \mathcal{O}(nk + n^2 \log n)$ time.

For the rest of the proof, let us fix a segment $r \in S$. Computing $T_r$ takes $\mathcal{O}(|E(\mathbb{G})| + |V(\mathbb{G})| \log |V(\mathbb{G})|) = \mathcal{O}(k + n \log n)$ time. For any segment $s \in S$, $s \neq r$, let $T_r[s]$ denote the path in $T_r$ from $r$ to $s$. We define $N_r(s) = N(\gamma(T_r[s]); a, b)$ and define $C_r(s)$ to be the child of $r$ in the path $T_r[s]$. See Fig. 3(a)–(b). The values $N_r(s)$, $s \in S$, can be computed in $\mathcal{O}(n)$ time using a BFS traversal of $T_r$: if $p_r(s)$ is the parent of $s$ in $T_r$, we can compute $N_r(s)$ from $N_r(p_r(s))$ in $\mathcal{O}(1)$ time using

$$N_r(s) \;=\; N_r(p_r(s)) + N(\gamma(p_r(p_r(s))p_r(s)s); a, b).$$

Similarly $C_r(s)$, $s \in S$, can be computed in $\mathcal{O}(n)$ time: we assign $C_r(s') = s'$ for all children $s'$ of $r$ and use that $C_r(s) = C_r(p_r(s))$ for any $s$ not adjacent to $r$.

For $ss' \in E(\mathbb{G}) \setminus E(T_r)$ we have that $N(\gamma(r, ss') : a, b)$ is equal to

$$N_r(s) + N(\gamma(p_r(s)ss'p_r(s')); a, b) - N_r(r, s') + N(\gamma(C_r(s')rC_r(s)); a, b).$$

See Fig. 3(b)–(c). (The negative sign comes from the reversal of $T_r[s]$.) Therefore, each $N(\gamma(r, ss'); a, b)$ can be computed in $\mathcal{O}(1)$ time from the values $N_r(s)$, $N_r(s')$, $C_r(s')$, $C_r(s)$. It follows that $E_r$ can be constructed in $\mathcal{O}(|E(\mathbb{G})|) = \mathcal{O}(k)$ time.

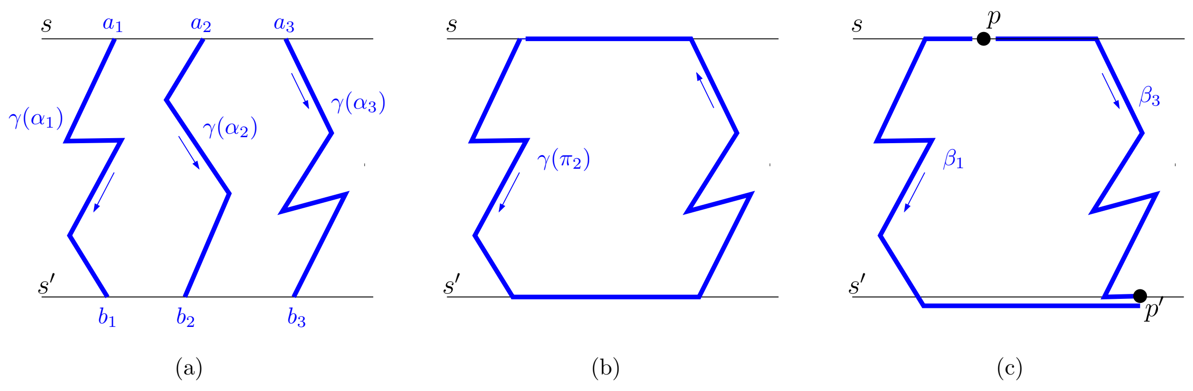


Figure 4: (a) Scenario in the proof of Lemma 3. (b) The polygonal path $\gamma(\pi_2)$. (c) The polygonal paths $\beta_1$ and $\beta_3$. (The bottom of $\beta_1$ lies on $s'$. We draw it outside because of the common part with $\beta_3$.)

The length of any cycle $\tau(T_r, e)$ can be computed in $\mathcal{O}(1)$ time per cycle in a similar fashion. For each vertex $s$, we store at $s$ its distance $d_{\mathbb{G}}(r, s)$ from the root $r$. We also construct a data structure for finding lowest common ancestor (lca) of two vertices in constant time. Such data structure can be constructed in $\mathcal{O}(n)$ time [BFC04, HT84]. The length of a cycle can then be recovered using

$$\text{len}_{\mathbb{G}}(\tau(T_r, ss')) = d_{\mathbb{G}}(r, s) + w(s) + w(s') + d_{\mathbb{G}}(r, s') - 2d_{\mathbb{G}}(r, \text{lca}(s, s')).$$

Equipped with this, we can in $\mathcal{O}(k)$ time compute

$$\min_{e \in E_r} \text{len}_{\mathbb{G}}(\tau(T_r, e)). \qquad \square$$

The following special case will be also relevant later on.

**Lemma 2.** *If the weights of the segments $S$ are 0 or 1, then the pair $(r^*, e^*)$ can be computed in $\mathcal{O}(nk + n^2)$ time.*

*Proof.* In this case, a shortest path tree $T_r$ can be computed in $\mathcal{O}(|E(\mathbb{G})| + |V(\mathbb{G})|) = \mathcal{O}(k+n)$ time because the edge weights of $\mathbb{G}$ are 0, 1, or 2. Using the approach described in the proof of Lemma 1 we spend $\mathcal{O}(k + n)$ per root $r \in S$, and thus spend $\mathcal{O}(nk + n^2)$ in total. $\square$

## 2.2 Correctness

Consider the set of closed walks

$$\Pi = \{\pi \mid \pi \text{ a closed walk in } \mathbb{G} \text{ with } N(\gamma(\pi); a, b) \neq 0\}.$$

We have the following property, known as 3-path condition.

**Lemma 3.** *Let $\alpha_0, \alpha_1, \alpha_2$ be 3 walks in $\mathbb{G}$ from $s$ to $s'$. For $i = 0, 1, 2$, let $\pi_i$ be the closed walk obtained by concatenating $\alpha_{i-1}$ and the reverse of $\alpha_{i+1}$, where indices are modulo 3. If one of the walks $\pi_0, \pi_1, \pi_2$ is in $\Pi$, then at least two of them are in $\Pi$.*

*Proof.* (This result is a consequence of the group structure for relative $\mathbb{Z}_2$-homology. We provide an elementary proof that avoids using homology.) For $i = 0, 1, 2$, let $a_i$ be the starting vertex of the polygonal path $\gamma(\alpha_i)$ and let $b_i$ be the ending vertex. The polygonal paths $\gamma(\alpha_0), \gamma(\alpha_1), \gamma(\alpha_2)$ start on $s$ and finish on $s'$. However, they may have different endpoints. See Fig. 4. To handle this, we choose a point $p$ on $s$ and a point $p'$ on $s'$, and define $\beta_i$ to be the polygonal path obtained by the concatenation of $pa_i$, $\gamma(\alpha_i)$, and $b_ip'$. A simple but tedious calculation shows that, using indices modulo 3,

$$N(\gamma(\pi_i); a, b) \quad = \quad N(\beta_{i-1}; a, b) - N(\beta_{i+1}; a, b).$$

Indeed, since

$$N(a_{i+1}a_{i-1}; a, b) = N(pa_{i-1}; a, b) + N(a_{i+1}p; a, b)$$

and

$$N(b_{i-1}b_{i+1}; a, b) = N(p'b_{i+1}; a, b) + N(b_{i-1}p'; a, b),$$

we have

$$\begin{aligned} N(\gamma(\pi_i); a, b) &= \quad N(\gamma(\alpha_{i-1}); a, b) + N(b_{i-1}b_{i+1}; a, b) \\ &\quad + N(\text{reverse}(\gamma(\alpha_{i+1})); a, b) + N(a_{i+1}a_{i-1}; a, b) \\ &= \quad N(\gamma(\alpha_{i-1}); a, b) + N(p'b_{i+1}; a, b) + N(b_{i-1}p'; a, b) \\ &\quad - N(\gamma(\alpha_{i+1}); a, b) + N(pa_{i-1}; a, b) + N(a_{i+1}p; a, b) \\ &= \quad N(pa_{i-1}; a, b) + N(\gamma(\alpha_{i-1}); a, b) + N(b_{i-1}p'; a, b) \\ &\quad - N(pa_{i+1}; a, b) - N(\gamma(\alpha_{i+1}); a, b) - N(b_{i+1}p'; a, b) \\ &= \quad N(\beta_{i-1}; a, b) - N(\beta_{i+1}; a, b). \end{aligned}$$

It follows that, using indices modulo 3,

$$\sum_{i=0}^{2} N(\gamma(\pi_i); a, b) \quad = \quad \sum_{i=0}^{2} \left(N(\beta_{i-1}; a, b) - N(\beta_{i+1}; a, b)\right) \quad = \quad 0.$$

Therefore, if $N(\gamma(\pi_i); a, b) \neq 0$ for some $i$, at least another cycle $\pi_j$, $j \neq i$, must have $N(\gamma(\pi_j); a, b) \neq 0$. □

When a family of closed walks satisfies the 3-path condition, there is a general method to find a shortest element in the family. The method is based on considering fundamental-cycles defined by shortest-path trees, which is precisely what our algorithm is doing specialized for the family $\Pi$. We thus obtain:

**Lemma 4.** *The cycle $\tau^*$ is a shortest element of* $\Pi$*.*

*Proof.* It is a consequence of the 3-path condition, that a shortest cycle in

$$\{\tau(T_r, e) \mid r \in S, e \in E(\mathbb{G}) \setminus E(T_r), \tau(T_r, e) \in \Pi\} \quad = \quad \{\tau(T_r, e) \mid (r, e) \in P\}$$

is a shortest cycle in $\Pi$. That is, the search for a shortest element in $\Pi$ can be restricted to cycles of the type $\tau(T_r, e)$. See Thomassen [Tho90] or the book by Mohar and Thomassen [MT01, Chapter 4] for the so-called fundamental cycle method. (The method is described for unweighted graphs but it also works for weighted graphs. See, for example, Cabello et al. [CdVL10] for the generalized case of weighted, directed graphs.) □

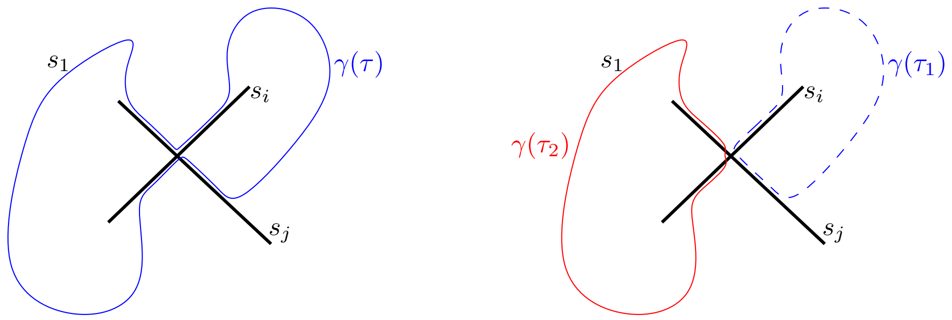


Figure 5: The polygonal paths defined by the cycles $\tau_1$ and $\tau_2$ from the cycle $\tau$ in Lemma 6.

The next step in our argument is showing that $\gamma^*$ is simple (without self-intersections) and separates $a$ and $b$. We will use the following characterization of which simple, closed polygonal paths separate $a$ and $b$.

**Lemma 5.** *For any simple, closed polygonal path $\gamma$ we have $|N(\gamma; a, b)| \leq 1$. Furthermore, $\gamma$ separates $a$ and $b$ if and only if $N(\gamma; a, b) = \pm 1$.*

*Proof.* Since $\gamma$ is simple, it defines an interior and an exterior by the Jordan curve theorem. The crossings between $\gamma$ and $\overline{ab}$, as we walk along $\overline{ab}$, alternate between left-to-right and right-to-left crossings because $\overline{ab}$ has pieces alternating in the interior and exterior of $\gamma$. Therefore $|N(\gamma; a, b)| \leq 1$.

Assume that $\gamma$ separates $a$ and $b$, so that one is in the interior of $\gamma$ and the other in the exterior. Then the segment $\overline{ab}$ crosses $\gamma$ an odd number of times, and it must be $|N(\gamma; a, b)| = 1$. Conversely, if $|N(\gamma; a, b)| = 1$, then the number of intersections between $\gamma$ and $\overline{ab}$ is odd, which implies that one of the points $a$ and $b$ is in the interior of $\gamma$ and the other in the exterior. □

We can now prove that $\gamma^*$ is simple using a standard uncrossing argument. Indeed, a self-crossing of $\gamma^*$ would imply that we can find a strictly shorter element in $\Pi$, which would contradict the property stated in Lemma 4.

**Lemma 6.** *The polygonal path $\gamma^*$ is simple and separates $a$ and $b$.*

*Proof.* Assume, for the sake of contradiction, that $\gamma^*$ is not simple. It is then possible to show the existence of two cycles $\tau_1$ and $\tau_2$ in $\mathbb{G}$ such that $\text{len}_{\mathbb{G}}(\tau_1) < \text{len}_{\mathbb{G}}(\tau^*)$, $\text{len}_{\mathbb{G}}(\tau_2) < \text{len}_{\mathbb{G}}(\tau^*)$, and $N(\gamma^*; a, b) = N(\gamma(\tau_1); a, b) + N(\gamma(\tau_2); a, b)$, as follows.

Let $s_0 s_1 s_2 \dots s_t$, with $s_t = s_0$, be the cycle $\tau^*$. Start walking along $\gamma^*$ from $s_0 \cap s_1$, until we find the first self-intersection, which is defined by segments $s_i$ and $s_j$, with $i < j$. Note that $2 \leq j - i$ because $s_i$ and $s_{i+1}$ cannot define a self-intersection of $\gamma^*$. Consider the cycles $\tau_1 = s_i s_{i+1} \dots s_j s_i$ and $\tau_2 = s_0 \dots s_i s_j \dots s_t$. See Fig. 5. Note that

$$N(\gamma^*; a, b) \quad = \quad N(\gamma(\tau_1); a, b) + N(\gamma(\tau_2); a, b)$$

because the polygonal paths $\gamma(\tau_1)$ and $\gamma(\tau_2)$ form a disjoint partition of $\gamma^*$, with orientations preserved. Moreover, because $j - i \geq 2$ and $\tau^*$ is a cycle, we have $\text{len}_{\mathbb{G}}(\tau_1) < \text{len}_{\mathbb{G}}(\tau^*)$ and $\text{len}_{\mathbb{G}}(\tau_2) < \text{len}_{\mathbb{G}}(\tau^*)$. This finishes the proof of existence of $\tau_1$ and $\tau_2$.

Because $\tau^* \in \Pi$ we have

$$0 \neq N(\gamma^*; a, b) = N(\gamma(\tau_1); a, b) + N(\gamma(\tau_2); a, b).$$

Therefore, $N(\gamma(\tau'); a, b) \neq 0$ for some $\tau' \in \{\tau_1, \tau_2\}$. Since $\text{len}_{\mathbb{G}}(\tau') < \text{len}_{\mathbb{G}}(\tau^*)$ and $N(\gamma(\tau'); a, b) \neq 0$, then $\tau' \in \Pi$. This contradicts the property that $\tau^*$ is a shortest cycle of $\Pi$ (Lemma 4). We conclude that $\gamma^*$ must be simple.

Since $\gamma^*$ is simple, $N(\gamma^*) \in \{-1, 0, +1\}$ by Lemma 5. Since $\tau^* \in \Pi$, then $N(\gamma^*) \neq 0$, which implies $N(\gamma^*) = \pm 1$. It then follows from Lemma 5 that $\gamma^*$ separates $a$ and $b$. $\square$

We can now prove the main theorem.

**Theorem 7.** *The weighted version of* 2-Cells-Separation *can be solved in $\mathcal{O}(nk+n^2 \log n)$ time, where $n$ is the number of input segments and $k$ is the number of pairs of segments that intersect.*

*Proof.* We use the algorithm described in Section 2.1. The algorithm returns a feasible solution because of Lemma 6: the cycle $\gamma^*$ separates $a$ and $b$ and is contained in $\bigcup S(\tau(T_{r^*}, e^*))$, therefore, the set $S(\tau(T_{r^*}, e^*)$ returned by the algorithm separates $a$ and $b$.

To see the optimality of the weight of $S(\tau^*)$, consider an optimal solution $S_* \subseteq S$. Assume that we run the algorithm on $S_*$. The algorithm would compute a cycle $\tau_*$ in the intersection graph of the segments $S_*$ and return $S(\tau_*) \subseteq S_*$. By Lemma 6, the polygonal path $\gamma(\tau_*)$ is simple and separates $a$ and $b$. Lemma 5 implies that $N(\gamma(\tau_*); a, b) = \pm 1 \neq 0$, and therefore $\tau_* \in \Pi$ (here $\Pi$ refers to the original problem, rather than the subproblem defined by input $S_*$).

For any cycle $\pi$ of $\mathbb{G}$ we have $\text{len}_{\mathbb{G}}(\pi) = 2|S(\pi)|$ because of the choice of the edge-weights in $\mathbb{G}$. Since $\tau^*$ is a shortest cycle in $\Pi$ by Lemma 4, we have

$$w(S(\tau^*)) = \tfrac{1}{2}\,\text{len}_{\mathbb{G}}(\tau^*) \leq \tfrac{1}{2}\,\text{len}_{\mathbb{G}}(\tau_*) = w(S(\tau_*)) \leq w(S_*).$$

It follows that $S(\tau^*)$ is a feasible solution whose weight is not larger than $w(S_*)$, and therefore it is optimal. The running time follows from Lemma 1. $\square$

**Corollary 8.** *The weighted version of* 2-Cells-Separation *in which the segments have weights* 0 *or* 1 *can be solved in $\mathcal{O}(n^2 + nk)$ time, where $n$ is the number of input segments and $k$ is the number of pairs of segments that intersect.*

*Proof.* In the proof of the previous theorem we use Lemma 2 instead of Lemma 1. $\square$

In the case where the segments of $S$ are unweighted, the points $a, b$ are inside a polygon $P$ with holes, and the $a$-$b$ path must be contained in the interior of $P$, the problem can be easily solved by assigning weight 0 to the edges $E(P)$ of the polygon $P$ and weight 1 to the segments in $S$. We can then apply Corollary 8 on $S \cup E(P)$, and obtain the following.

**Corollary 9.** *The restricted* 2-Cells-Separation *problem in a polygon with holes can be solved in $\mathcal{O}(n^2 + nk)$ time, where $n$ is the total size of the input and $k$ is the number of pairs of segments in $S$ that intersect.*

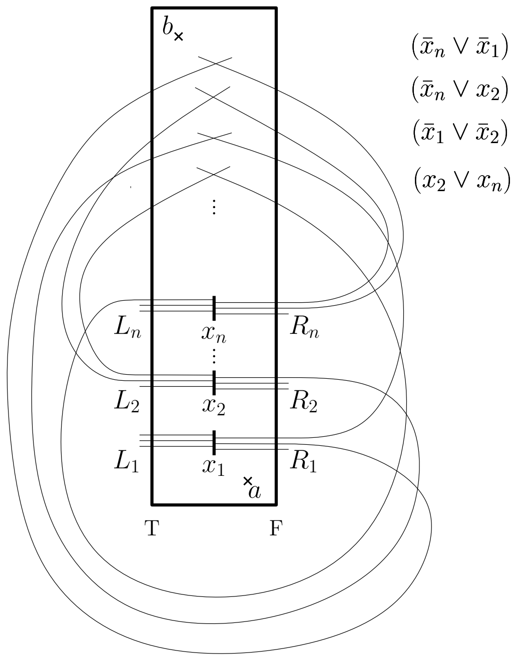


Figure 6: Idea of the construction with curved segments.

## 3 Connecting two cells

We show that 2-Cells-Connection is NP-hard and APX-hard by a reduction from Exact-Max-2-Sat, a well studied NP-complete and APX-complete problem(c.f. [Hås01]): Given a propositional CNF formula $\Phi$ with $m$ clauses on $n$ variables and exactly two variables per clause, decide whether there exists a truth assignment that satisfies at least $k$ clauses, for a given $k \in \mathbb{N}$, $k \leq m$. Let $x_1, \dots, x_n$ be the variables of $\Phi$, $\ell_i$ be the number of appearances of variable $x_i$ in $\Phi$, and $\ell = \sum_i \ell_i$; since each clause contains exactly 2 variables, $\ell = 2m$. The maximum number of satisfiable clauses is denoted by opt($\Phi$). Using $\Phi$ we construct an instance consisting of a set of segments $S = S(\Phi)$ and two points $a = a(\Phi)$ and $b = b(\Phi)$ as follows.

Abusing the terminology slightly, the term *segment* will refer to a set of identical single segments stacked on top of each other. The cardinality of the set is the *weight* of the segment. Either all or none of the single segments in the set can be crossed by a path. There are two different types of segments, $\tau_1$, and $\tau_\infty$, according to their weight. Segments of type $\tau_1$ have weight 1 (light or single segments), while segments of type $\tau_\infty$ have weight $20m$ (heavy segments). The weight of heavy segments is chosen so that they are never crossed by an optimal $a$-$b$ path.

We first provide an informal, high-level description of the construction that uses *curved* segments. Later on, each curved segment will by replaced by a collection of straight-line segments in an appropriate manner. See Fig. 6. We have a rectangle $R_\infty$ made of heavy segments, with point $a$ at a lower corner and $b$ at an upper corner. For each variable $x_i$ we add a small vertical segment of type $\tau_\infty$ in the lower half of $R_\infty$. From the segment we place $\ell_i$ horizontal light segments, denoted by $R_i$, going to the right and $\ell_i$ horizontal light segments, denoted by $L_i$, going to the left until they reach the outside of $R_\infty$. Roughly speaking, (things

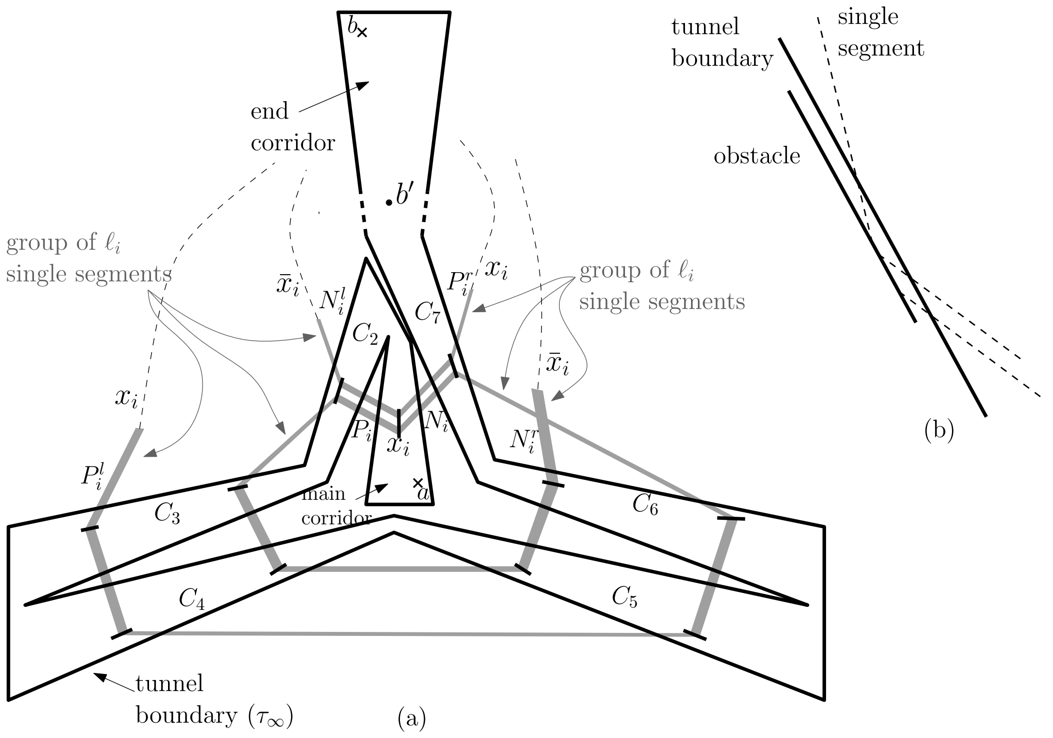


Figure 7: (a) Tunnel and variable chain. Each gray trapezoid represents a piece with $\ell_i$ parallel segments. (b) Part of a chain piece close to the tunnel.

are slightly more complicated) an optimal $a$-$b$ path will have to choose for each $x_i$ whether it crosses all segments in $L_i$, encoding the assignment $x_i = \mathrm{T}$, or all segments in $R_i$, encoding the assignment $x_i = \mathrm{F}$. Consider a clause like $x_2 \vee x_n$, where both literals are positive. We prolong one of the segments of $L_2$ and one of the segments of $L_n$ with a curved segment so that they cross again inside $R_\infty$ (upper half) in such a way that an $a$-$b$ path inside $R_\infty$ must cross one of the prolongations, and one is enough; see Fig. 6, where one of the prolongations passes below $R_\infty$. A clause like $\bar{x}_n \vee x_2$ is represented using prolongations of one segment from $L_2$ and one segment of $R_n$. The other types of clauses are symmetric. For each clause we always prolong different segments; since $L_i$ and $R_i$ have $\ell_i$ segments, there is always some segment that can be prolonged. It will then be possible to argue that the optimal $a$-$b$ path has cost $\ell + (m - \mathrm{opt}(\Phi))$. We do not provide a careful argument of this here since we will need it later for a most complicated scenario. This finishes the informal description of the idea.

We now describe in detail the construction with straight-line segments. First, we construct a polygon, called the *tunnel*, with heavy boundary segments of type $\tau_\infty$; see Fig. 7(a). The tunnel has a 'zig-zag' shape and can be seen as having 8 corridors, $C_1, \ldots, C_8$. It starts with $C_1$, the *main* corridor (at the center of the figure), which contains point $a$, then it turns left to $C_2$, then right, etc., gradually turning around to $C_7$ and then to the *end* corridor $C_8$ (at the top). The latter contains point $b$. To facilitate the discussion, we place a point $b'$ in the tunnel where the transition from $C_7$ to the end corridor occurs. The tunnel has a total weight of $21 \cdot 20m = \mathcal{O}(m)$. The rest of the construction will force any $a$-$b$ path of some particular cost (to be given shortly) to stay always in the interior of the tunnel.

Each variable $x_i$ of $\Phi$ is represented by a collection of 16 pieces, which form a chain-like structure. Each *piece* is a group of $\ell_i$ nearly-parallel single segments whose ends are either outside the tunnel or lie on 'short' heavy segments of type $\tau_\infty$ in the interior of the tunnel, referred to as *obstacles.* For each variable, there is one obstacle in each of the corridors $C_1$, $C_2$, $C_7$ and there are two obstacles in each of the corridors $C_3$, $C_4$, $C_5$, and $C_6$. See Fig. 7(a), where we represent each piece by a light gray trapezoid and each obstacle by a bold, short segment. Pieces always contain a part outside the tunnel. The exact description of the structure is cumbersome; we refer the reader to the figures. The obstacle in $C_2$ contains the extremes of four pieces: two pieces, called $P_i$, go to the obstacle in the main corridor, one goes to an obstacle in $C_3$, and the fourth piece, which we call $N_i^l$ goes outside the tunnel. Symmetrically, the obstacle in $C_7$ contains the extremes of four pieces: two pieces, called $N_i$, go to the main corridor, one goes to the corridor $C_6$, and one, which we call $P_i^r$ goes outside the tunnel. We add pieces connecting the obstacles in $C_3$ and $C_4$, the obstacles in $C_4$ and $C_5$, and the obstacles in $C_5$ and $C_6$. From the obstacle in $C_3$ that currently has one piece we add another piece, which we call $P_i^l$ and whose other extreme is outside the tunnel. From the obstacle in $C_6$ that currently has one piece we add another piece, which we call $N_i^r$, whose other extreme is outside the tunnel.

The obstacles and the pieces of all variables should satisfy some conditions: obstacles should be disjoint, pieces can touch only the obstacles at their extremes, and pieces may cross only outside the tunnel. See Fig. 8. Some of the single segments of $P_i^r$, $P_i^l$, $N_i^r$, $N_i^l$ will be prolonged and rotated slightly to encode the clauses. For this, we will need that the line supporting a segment from $P_i^r \cup N_i^r$ intersects inside the end corridor the line supporting a segment from $P_j^l \cup N_j^l$. This can be achieved by stretching the end corridor sufficiently and placing the obstacles of $C_2$ and $C_7$ close to the tunnel boundary; see Fig. 7(b).

For each clause of $\Phi$ we prolong two segments of $P_i^r \cup P_i^l \cup N_i^r \cup N_i^l$ as follows; see Fig. 8 for an example of the overall construction, where prolongations are shown by dashed lines. Each segment corresponds to some literal $x_i$ or $\bar{x}_i$ in the clause: in the first case the segment comes from either $P_i^r$ or $P_i^l$, while in the second one it comes from either $N_i^r$ or $N_i^l$. For the construction, these choices for each clause can be made arbitrarily, provided that one segment intersects the tunnel from the left side and the other one from the right. These segments are prolonged until their intersection point inside the end corridor. For each clause, two different segments are prolonged. Since the pieces corresponding to variable $x_i$ have $\ell_i$ segments, there is always some segment available. Segments corresponding to different clauses may intersect only outside the tunnel; this is ensured by rotating the segments slightly around the endpoint lying in the obstacle. In this way, the end corridor is obstructed by $m$ pairs of intersecting segments such that any path from the intermediate point $b'$ to point $b$ staying inside the tunnel must intersect at least one segment from each pair.

The following lemma establishes the correctness of the reduction.

**Lemma 10.** *There is an a-b path of cost at most $8\ell + k$, where $1 \le k \le m$, if and only if there is a truth assignment satisfying at least $(m - k)$ of the clauses.*

*Proof.* We denote by $S_i$ the set of segments in the pieces corresponding to the variable $x_i$. We denote by $S_i^{\mathrm{T}}$ the segments in the pieces of $P_i$, the piece connecting $C_7$ to $C_6$, the piece $P_i^r$, and so on in an alternating manner along the chain structure. Note that $S_i^{\mathrm{T}}$ contains $P_i$, $P_i^l$ and $P_i^r$. We denote by $S_i^{\mathrm{F}}$ the segments $S_i \setminus S_i^{\mathrm{T}}$. Note that $S_i^{\mathrm{F}}$ contains $N_i$, $N_i^l$ and $N_i^r$. See Fig. 9. Each of the sets $S_i^{\mathrm{T}}$ and $S_i^{\mathrm{F}}$ contains $8\ell_i$ segments. Inside the tunnel there is an

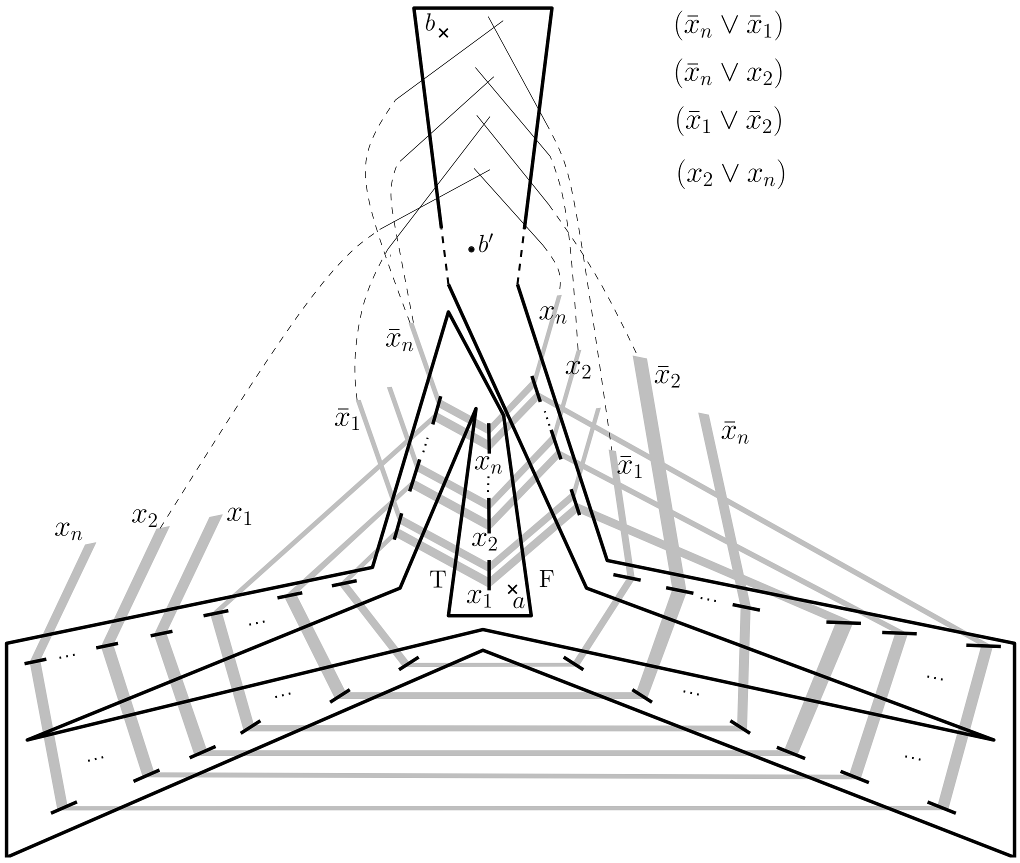
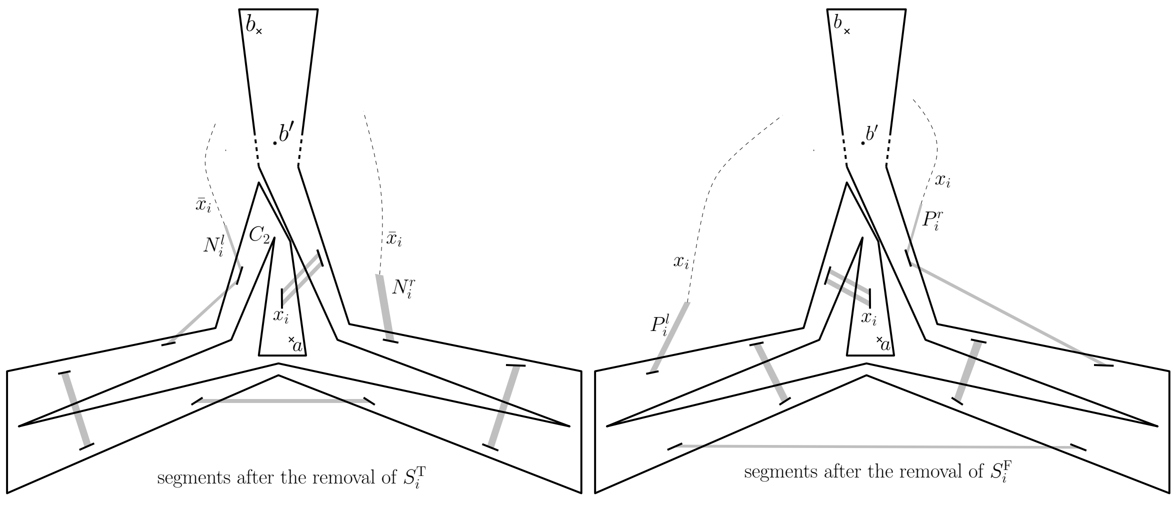


Figure 8: Example of overall construction.

$a$-$b'$ path disjoint from $S_i^{\mathrm{T}}$ and there is another $a$-$b'$ path disjoint from $S_i^{\mathrm{F}}$. We also denote by $T_j$ the two segments used for clause $C_j$ of $\Phi$.

Consider a truth assignment $\{x_i = b_i\}$, where each $b_i \in \{\mathrm{T}, \mathrm{F}\}$, satisfying at least $(m-k)$ clauses. We construct a subset of segments $S'$ where we include the set $S_i^{b_i}$, for each variable $x_i$, and a segment of $T_j$, for each clause $C_j$ that is not satisfied by the truth assignment. Since $|S_i^{b_i}| = 8\ell_i$, the set $S'$ contains at most $8\ell + k$ segments. The removal of $S'$ leaves the points $a$ and $b'$ in the same cell of the arrangement. Equivalently, there is an $a$-$b'$ path inside the tunnel that crosses only segments from $S'$. If a clause $C_j$ of $\Phi$ is satisfied by the truth assignment, then at least one of the segments in $T_j$ is included in $S_i^{b_i} \subset S'$. If a clause $C_j$ is not satisfied, then one of the segments $T_j$ is included in $S'$ by construction. Thus, for each clause $C_j$ we have $T_j \cap S' \neq \emptyset$. It follows that $b'$ and $b$ are in the same cell after the removal of $S'$.

Conversely, note first that any $a$-$b$ path with cost at most $8\ell + k \leq 16m + m = 17m$ cannot intersect the tunnel boundary or an obstacle because segments of type $\tau_\infty$ have weight $20m$. Let $S'$ be the set of segments crossed by the path. If $P_i \subset S'$, then we define $b_i = \mathrm{T}$; otherwise, we define $b_i = \mathrm{F}$. Note that when $P_i \not\subset S'$, then $N_i \subset S'$ because the $a$-$b$ path is inside the tunnel. (However it may be $N_i \cup P_i \subset S'$, so the assignment of $b_i$ is not symmetric.) We next argue that the truth assignment $\{x_i = b_i\}$ satisfies at least $(m-k)$ clauses.



$b$
$b'$
$\bar{x}_i$
$N_i^l$
$C_2$
$\bar{x}_i$
$N_i^r$
$x_i$
$a$
segments after the removal of $S_i^{\mathrm{T}}$
$b$
$b'$
$x_i$
$P_i^r$
$x_i$
$P_i^l$
$x_i$
$a$
segments after the removal of $S_i^{\mathrm{F}}$

Figure 9: Removal of $S_i^{\mathrm{T}}$ (left) and $S_i^{\mathrm{F}}$ (right).

Consider the case when $P_i \subset S'$. Inspection shows that

$$|S' \cap S_i| \geq 8\ell_i + |S' \cap (N_i^l \cup N_i^r)|.$$

Indeed, after the removal of $P_i \cup N_i^l \cup N_i^r$ any path from $a$ to $b'$ must still cross at least 6 pieces. Similarly, inspection shows that when $N_i \subset S'$ we have

$$|S' \cap S_i| \geq 8\ell_i + |S' \cap (P_i^l \cup P_i^r)|.$$

Let $A_i = N_i^l \cup N_i^r$ if $b_i = \mathrm{T}$ and $A_i = P_i^l \cup P_i^r$ if $b_i = \mathrm{F}$. The previous cases can be summarized as

$$|S' \cap S_i| \geq 8\ell_i + |S' \cap A_i|.$$

We further define

$$Y = \bigcup_i (S' \cap A_i).$$

For each clause $C_j$ we have $S' \cap T_j \neq \emptyset$ by construction, as otherwise $a$ and $b$ cannot be in the same cell of $S \setminus S'$. If $C_j$ is not satisfied by the truth assignment $\{x_i = b_i\}$, then it must be $(S' \cap T_j) \subset S' \cap A_k$ for some variable $x_k$ in $C_j$. This means that $T_j \cap Y \neq \emptyset$. Since the sets $T_j$ are disjoint by construction, the number of unsatisfied clauses is bounded by $|Y|$. Using that

$$8\ell + k \ = \ |S'| \ = \ \sum_{i=1}^{n} |S' \cap S_i| \ \geq \ \sum_{i=1}^{n} (8\ell_i + |S' \cap A_i|) \ = \ 8\ell + \sum_{i=1}^{n} |S' \cap A_i|,$$

we obtain

$$\sum_{i=1}^{n} |S' \cap A_i| \leq k.$$

Therefore, the total number of clauses with value F is bounded by

$$|Y| \ = \ \sum_i |S' \cap A_i| \ \leq \ k. \qquad \square$$

The construction can be easily modified by replacing every heavy segment with a set of $20m$ distinct parallel single segments such that every single segment in $S$ that originally intersected the heavy segment now intersects all the segments in the new set and such that no three segments have a point in common. We have the following:

**Theorem 11.** 2-Cells-Connection *is NP-hard and APX-hard even when no three segments intersect at a point.*

*Proof.* NP-hardness follows form Lemma 10 and the fact that the reduction produces $\mathcal{O}(nm)$ segments, whose coordinates can be bounded by a polynomial in $(n+m)$. APX-hardness follows from the fact that the reduction is approximation-preserving, as we now show.

First, since there is always an assignment that satisfies at least $3m/4$ clauses, we have that $m \leq (4/3)\mathrm{opt}(\Phi)$. Recall that an optimal $a$-$b$ path costs $8\ell + (m - \mathrm{opt}(\Phi))$, where $\ell = 2m$. A polynomial-time $c$-approximation algorithm ($c > 1$) for the problem would give a path that costs at most

$$\begin{aligned}
c(8\ell + (m - \mathrm{opt}(\Phi))) &= c(17m + \mathrm{opt}(\Phi)) \\
&= 17m - c\,\mathrm{opt}(\Phi) + 17(c-1)m \\
&\leq 17m - c\,\mathrm{opt}(\Phi) + 17(c-1)(4/3)\mathrm{opt}(\Phi) \\
&= 17m - \mathrm{opt}(\Phi)(68/3 - (65/3)c) \\
&= 16m + \Big[m - \mathrm{opt}(\Phi)(68/3 - 65c/3)\Big]
\end{aligned}$$

and, by Lemma 10, a truth assignment that satisfies at least $\mathrm{opt}(\Phi)(68/3 - 65c/3)$ clauses. However, Exact-Max-2-SAT cannot be approximated above $21/22$ [Hås01], which implies that $c$ must be larger than $(68/65 - 63/(22 \cdot 65)) \approx 1.002097\ldots$ (A slightly better inapproximability result can be obtained using the better bounds that rely on the unique games conjecture [KKMO07].) □

We can reduce 2-Cells-Connection to the minimum color path problem (MCP): Given a graph G with colored (or labeled) edges and two of its vertices, find a path between the vertices that uses the minimum possible number of colors. We color the edges of the dual graph $G$ of $\mathcal{A}(S)$ as follows: two edges of $G$ get the same color if and only if their corresponding edges in $\mathcal{A}(S)$ lie on the same segment of $S$. Then, finding an $a$-$b$ path of cost $k$ in $\mathcal{A}(S)$ amounts to finding a $k$-color path in $G$ between the two cells which $a$, $b$ lie in.

However, MCP is NP-hard [BLWZ05] and W[1]-hard [FGI10] (with respect to solution size) even for planar graphs, it has a polynomial-time $\mathcal{O}(\sqrt{n})$-approximation algorithm and is non-approximable within any polylogarithmic factor [HMS07].

## 4 Tractable cases for connecting two cells

We now describe two special cases where 2-Cells-Connection is tractable. First, we consider the case where the input segments have few crossings, in a sense that is specified below. Then, we return to the special case where we have a polygon and provide an algorithm that takes polynomial time when the number of holes in the polygon is constant.

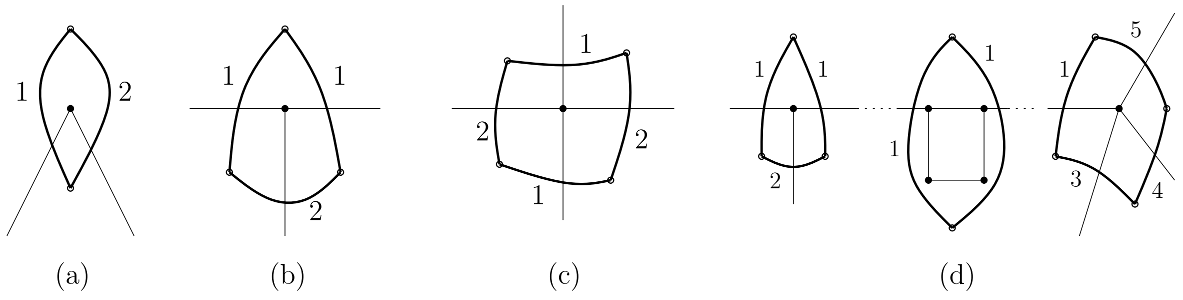


Figure 10: Examples of intersections in $\mathcal{A}(S)$ and colored edges in $G$.

## 4.1 Segments crossings.

Without loss of generality, we assume that every segment in $S$ intersects at least two other segments and that both endpoints of a segment are intersection points. We say that two segments cross if and only if they intersect at a point that is interior to both segments (a segment crossing).

Consider the colored dual graph $G$ of $\mathcal{A}(S)$ as defined after Theorem 11. A face of $G$ (except the outer one) corresponds to a point of intersection of some $r \geq 2$ segments and has $r$ colors and, depending on the type of intersection, from $r$ to $2r$ edges. For example, for $r = 2$ we can get two multiple edges, a triangle, or a quadrilateral, with two distinct colors. See Fig. 10(a)-(c), where the colors are given as labels.

When any three segments may intersect only at a common endpoint and no two segments cross, $G$ can only have multiple edges (possible all with the same color), bi-chromatic triangles, and arbitrary large faces where all edges have different colors; See Fig. 10(d) for an example. In this case, since two segments can intersect only at one point, each color induces a connected subgraph of $G$, in fact a tree (where all but one multiple edges with the same color can be deleted) for there can be no monochromatic cycle in $G$. Then, 2-Cells-Connection reduces to a simple shortest path computation between the cells containing $a$ and $b$ in the (uncolored) graph resulting from $G$ by completing each monochromatic tree into a clique. By contrast, note that All-Cells-Connection is still NP-hard for this special case; see Section 5.

Generalizing this, if we allow $k$ segment crossings, we can easily reduce the problem to $2^{\mathcal{O}(k)}$ shortest path problems as follows. Let $C \subseteq S$ be the set of the (at most $2k$) segments participating in these crossings. For a fixed subset $C'$ of $C$, we first contract every edge of $G$ corresponding to a segment in $C'$, effectively putting all segments of $C'$ into the solution. Then, we delete every edge corresponding to a segment in $C \setminus C'$ that still participates in a crossing, i.e., we exclude all crossing segments of $C \setminus C'$ from the solution. In the resulting (possibly disconnected) graph $G'$, each of the remaining colors induces again a monochromatic subtree, thus we can compute a shortest path as before and add $C'$ to the solution. Finally, we return a minimum size solution set over all $2^{\mathcal{O}(k)}$ possible subsets $C'$. Thus, we have just

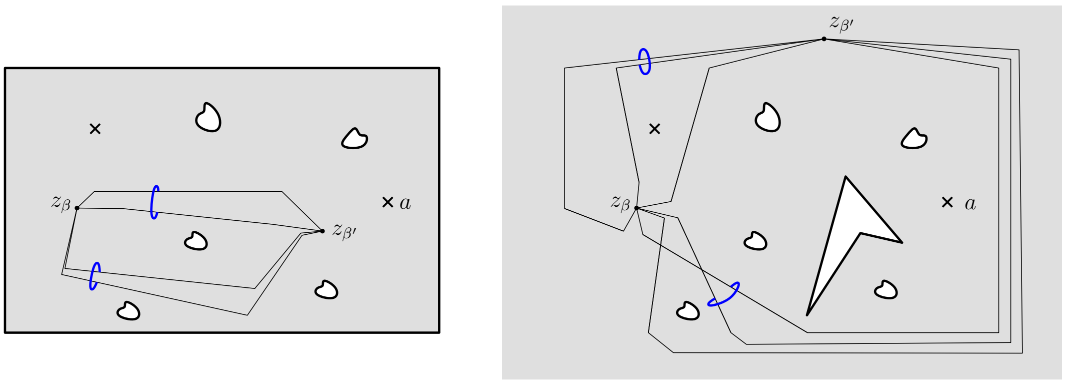


Figure 11: Some of the curves $\gamma_s$ arising from Fig. 1, after a small perturbation, and the resulting clusters. In the left case, $\beta$ and $\beta'$ are boundaries of holes, while in the right case $\beta'$ is the exterior boundary.

proved the following:

**Theorem 12.** 2-Cells-Connection *is fixed-parameter tractable with respect to the number of segment crossings if any three segments may intersect only at a common endpoint.*

## 4.2 Polygon with holes.

Let $P$ be a polygon with $h$ holes and $S$ be a set of segments lying inside $P$ with their endpoints on its boundary; see Fig. 1. We use $n$ as a bound for the number of vertices of $P$ and segments in $S$. We consider the restricted 2-Cells-Connection problem where the $a$-$b$ path may not cross the boundary of $P$. This version is also NP-hard by a simple reduction from the general one: place a large polygon enclosing all the segments and add a hole at the endpoint of each segment. We assume for simplicity that $a$ and $b$ are in the interior of $P$.

A boundary component of $P$ may be the exterior boundary or the boundary of a hole. For each boundary component $\beta$ of $P$, let $C_\beta$ be the connected component of $\mathbb{R}^2 \setminus P$ that has $\beta$ as boundary, and let $z_\beta$ be an arbitrary, fixed point in the interior of $C_\beta$. If $\beta$ is the exterior boundary, then $C_\beta$ is unbounded.

Let $\beta$ and $\beta'$ be two boundary components of $P$; it may be that $\beta = \beta'$. Let $S_{\beta,\beta'}$ be the subset of segments from $S$ with one endpoint in $\beta$ and another endpoint in $\beta'$. We partition $S_{\beta,\beta'}$ into clusters, as follows. Consider the set $X_{\beta,\beta'}$ obtained from $P \setminus \{a, b\}$ by adding $C_\beta$ and $C_{\beta'}$. Note that $a$ and $b$ are holes in $X_{\beta,\beta'}$. For each segment $s = \overline{pq} \in S_{\beta,\beta'}$, with $p \in \beta$ and $q \in \beta'$, we define the following curve $\gamma_s$: follow a shortest path in $C_\beta$ from $z_\beta$ to $p$, then follow $\overline{pq}$, and then follow a shortest path in $C_{\beta'}$ from $q$ to $z_{\beta'}$. See Fig. 11. We say that segments $s$ and $s'$ from $S_{\beta,\beta'}$ are *$a$-$b$ equivalent* if $\gamma_s$ and $\gamma_{s'}$ are homotopic paths in $X_{\beta,\beta'}$. Since being homotopic is an equivalence relation (reflexive, symmetric, transitive), being $a$-$b$ equivalent is also an equivalence relation in $S_{\beta,\beta'}$. Therefore, we can make equivalence classes, which we call *clusters*. The following two results provide key properties of the clusters.

**Lemma 13.** *$S_{\beta,\beta'}$ is partitioned into $\mathcal{O}(h^2)$ clusters. Such partition can be computed in $\mathcal{O}(hn \log n)$ time.*

*Proof.* Let $\Gamma_{\beta,\beta'}$ be the set of curves $\gamma_s$ over all segments $s \in S_{\beta,\beta'}$. Note that two curves $\gamma_s$ and $\gamma_{s'}$ of $\Gamma_{\beta,\beta'}$ may cross only once, and they do so along $s$ and $s'$. With a small perturbation

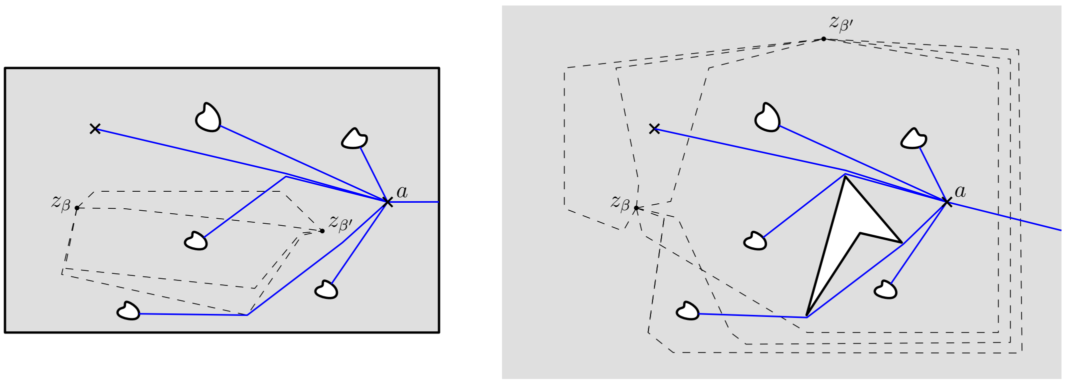


Figure 12: The curves $\Sigma$ in solid and $\Gamma_{\beta,\beta'}$ in dashed style for the example of Fig. 11, after a small perturbation.

of the curves in $\Gamma_{\beta,\beta'}$ we may assume that $\gamma_s$ and $\gamma_{s'}$ are either disjoint or cross at $s \cap s'$. (We do not actually use that $\gamma_s$ contains shortest paths inside $C_\beta$ and $C_{\beta'}$ besides for this property of non-crossing curves inside $C_\beta$ and $C_{\beta'}$.)

We now describe a simple criteria using crossing sequences to decide when two segments of $S_{\beta,\beta'}$ are $a$-$b$ equivalent. We take a set $\Sigma$ of non-crossing paths in $X_{\beta,\beta'}$ that have the following property: cutting $X_{\beta,\beta'}$ along the curves of $\sigma$ removes all holes. Such set $\Sigma$ has a tree-like structure and can be constructed as follows. For each boundary $\alpha$ of $P$, distinct from $\beta$ and $\beta'$, we add to $\Sigma$ the shortest path in $P$ between $a$ and $\alpha$. We add to $\Sigma$ the shortest path in $P$ between $a$ and $b$. Finally, if $\beta$ or $\beta'$ is the exterior boundary of $P$, we add to $\Sigma$ a shortest path from $a$ to a point that is very far in $P$ union the the outer face. In total, $\Sigma$ has $O(h)$ polygonal paths in $X_{\beta,\beta'}$. Note that the curves in $\Sigma$ are non-crossing and a small perturbation makes them disjoint, except at the common endpoint $a$. See Fig. 12. Each curve $\sigma \in \Sigma$ is simple and has two sides. We arbitrarily choose one of them as the right side and the other as the left side. We use $\sigma_1, \dots, \sigma_k$ to denote the curves of $\Sigma$.

To each path $\gamma$ in $\Gamma_{\beta,\beta'}$ we associate a crossing sequence $w(\gamma)$ as follows. We start with the empty word and walk along $\gamma$. When $\gamma$ crosses an arc $\sigma_i \in \Sigma$ from left-to-right we append $\sigma_i^{\rightarrow}$ to the word, and when $\gamma$ crosses $\sigma_i$ from right-to-left we append $\sigma_i^{\leftarrow}$ to the word. From the crossing sequence $w(\gamma)$ we can obtain the *reduced crossing sequence* $w^R(\gamma)$: we iteratively remove contiguous appearances of $\sigma_i^{\rightarrow}$ and $\sigma_i^{\leftarrow}$, for any $i$. For example, from the crossing sequence $\sigma_1^{\rightarrow}\sigma_2^{\leftarrow}\sigma_3^{\rightarrow}\sigma_3^{\leftarrow}\sigma_2^{\rightarrow}$ we obtain the reduced crossing sequence $\sigma_1^{\rightarrow}$. A consequence of using $\{\sigma_i\}$ to construct the so-called universal cover is the following characterization: the curves $\gamma_s$ and $\gamma_{s'}$ are homotopic in $X_{\beta,\beta'}$ if and only if the curves $\gamma_s$ and $\gamma_{s'}$ have the same reduced crossing sequence. See for example [CLMS04]. We conclude that $s$ and $s'$ from $S_{\beta,\beta'}$ are $a$-$b$ equivalent if and only if $w^R(\gamma_s) = w^R(\gamma_{s'})$.

The union of $\Sigma$ and $\Gamma_{\beta,\beta'}$ forms a family of pseudosegments: any two of them crosses at most once. Indeed, by construction different curves can only cross in $P$, but inside $P$ all those curves are shortest paths, and thus can cross at most once. Furthermore, the segments $\Sigma$ do not cross by construction and the curves of $\Gamma_{\beta,\beta'}$ have common endpoints. Mount [Mou90, Theorem 1.1] has shown that in such case the curves in $\Gamma_{\beta,\beta'}$ define at most $\mathcal{O}(|\Sigma|^2) = \mathcal{O}(h^2)$ distinct crossing sequences. Therefore, there are at most $\mathcal{O}(h^2)$ homotopy classes defined by the curves in $\Gamma_{\beta,\beta'}$, and $S_{\beta,\beta'}$ defines $\mathcal{O}(h^2)$ clusters.

The procedure we have described is constructive: we have to compute $O(h)$ shortest paths

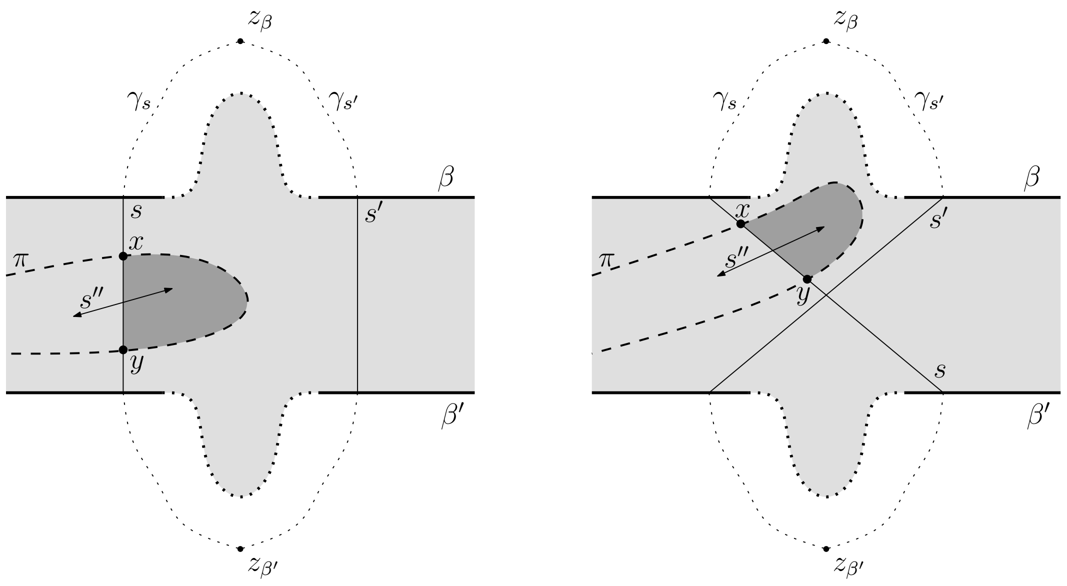


Figure 13: Figure for the proof of Lemma 14. Left: case when $s$ and $s'$ are disjoint. Right: case when $s$ and $s'$ intersect. In both cases, the darker gray region represents the topological disk defined by $\pi[x, y]$ and $\overline{xy}$.

in $P$ to obtain the curves of $\Sigma$, and then, for each segment $s \in S_{\beta,\beta'}$, we have to compute the corresponding crossing sequence. Such crossing sequence is already reduced. Note that for computing the crossing sequence of $\gamma_s$ we never have to construct $\gamma_s$ itself because all crossings occur along $s$. This can be done in $\mathcal{O}(hn \log n)$ time using algorithms for shortest paths in polygonal domains [HS99] and data structures for ray-shooting among the segments of $\Sigma$ [CEG+94]. □

**Lemma 14.** *For each cluster, either all or none of the segments in the cluster are crossed by a minimum-cost a-b path.*

*Proof.* Let $s$ and $s'$ be two $a$-$b$ equivalent segments from $S_{\beta,\beta'}$. This implies that $\gamma_s$ and $\gamma_{s'}$ are homotopic in $X_{\beta,\beta'}$. Therefore, the path $\gamma$ obtained by concatenating $\gamma_s$ and the reversal of $\gamma_{s'}$ is contractible in $X_{\beta,\beta'}$.

Let $\pi$ be a minimum-cost path between $a$ and $b$ that crosses $s$ but does not cross $s'$. We will reach a contradiction. We take $\pi$ that minimizes the total number of crossings with $s$. We may assume that $\pi$ is simple and disjoint from $\beta, \beta'$. We use $\pi[x, y]$ to denote the subpath of $\pi$ between points $x$ and $y$ of $\pi$. We distinguish two cases:

- $s$ and $s'$ do not intersect. In this case, the curve $\gamma$ is simple and contractible in $X_{\beta,\beta'}$. It follows that $\gamma$ bounds a topological disk $D_\gamma$ in $X_{\beta,\beta'}$. By hypothesis, $\pi$ crosses the part of the boundary of $D_\gamma$ defined by $s$ but not $s'$. Therefore, $\pi$ must cross at least twice along $s$. Let $x$ and $y$ be two consecutive crossings of $\pi$ and $s$ as we walk along $\pi$. See Fig. 13 left. Consider the path $\pi'$ that replaces $\pi[x, y]$ by the segment $\overline{xy}$. Any segment $s''$ crossing $s$ along $\overline{xy}$ crosses also $\pi$ because $\pi[x, y]$ and $\overline{xy}$ define a disk. Therefore $\pi'$ crosses no more segments than $\pi$ and crosses $s$ twice less than $\pi$. Thus, we reach a contradiction. (If $\pi'$ is not simple we can take a simple path contained in $\pi'$.)

- $s$ and $s'$ intersect. In this case, the curve $\gamma$ in $X_{\beta,\beta'}$ has precisely one crossing. Let $\gamma'$ and $\gamma''$ be the two simple loops obtained by splitting $\gamma$ at its unique crossing. It must be that $\gamma'$ and $\gamma''$ are contractible, as otherwise $\gamma$ would not be contractible. See Fig. 13 right. Therefore, we obtain two topological disks $D_{\gamma'}$ and $D_{\gamma''}$, one bounded by $\gamma'$ and another by $\gamma''$. The path $\pi$ must cross the boundary of $D_{\gamma'}$ or $D_{\gamma''}$, and the same argument than in the previous item leads to a contradiction. □

A minimum-cost $a$-$b$ path can now be found by testing all possible cluster subsets, that is, $2^{\mathcal{O}(h^4)}$ possibilities.

**Theorem 15.** *The restricted* 2-Cells-Connection *problem in a polygon with $h$ holes and $n$ segments can be found in $2^{\mathcal{O}(h^4)}$ polylog $n$ time.*

*Proof.* We classify the segments of $S$ into $\mathcal{O}(h^4)$ clusters using Lemma 13. This takes $\mathcal{O}(h^3 n \log n)$ time. Because of Lemma 14, we know that either all or none of the segments in a cluster are crossed by an optimal $a$-$b$ path. Each subset of the clusters defines a set of segments $S'$, and we can test whether $S'$ separates $a$ and $b$ in $\mathcal{O}(n \operatorname{polylog} n)$ time [GSS89, dBDS95]. □

## 5 Connecting all cells

We show that All-Cells-Connection is NP-hard by a reduction from the NP-hard problem of feedback vertex set (FVS) in planar graphs (c.f. [Vaz01]): Given a planar graph $G$, find a minimum-size set of vertices $X$ such that $G - X$ is acyclic.

First, we subdivide every edge of $G$ obtaining a planar bipartite graph $G'$. It is clear that $G'$ has a feedback vertex set of size $k$ if and only if $G$ has one. Next, we use the result by de Fraysseix et al. [dFOP91] (see also Hartman et al. [HNZ91]), which states that every planar bipartite graph is the intersection graph of horizontal and vertical segments, where no two of them cross (intersect at a common interior point). Let $S$ be the set of segments whose intersection graph is $G'$; it can be constructed in polynomial time. Since $G'$ has no triangles, no three segments of $S$ intersect at a point. Then, observe that all cells in $\mathcal{A}(S)$ become connected by removing $k$ segments if and only if $G'$ has a feedback vertex set of size $k$. Therefore we have:

**Theorem 16.** All-Cells-Connection *in NP-hard even if no three segments intersect at a point and there are no segment crossings.*

It is also easy to see that if no three segments intersect at a point a $k$-size solution to All-Cells-Connection corresponds to a $k$-size solution of FVS in the intersection graph of the input segments. For general graphs, FVS is fixed-parameter tractable when parameterized with the size of the solution [CFL+08], and has a polynomial-time 2-approximation algorithm [Vaz01]. We thus obtain the following:

**Corollary 17.** *When no three segments intersect at a point,* All-Cells-Connection *is fixed-parameter tractable with respect to the size of the solution and has a polynomial-time 2-approximation algorithm.*

## Acknowledgments:

We would like to thank Primož Škraba for bringing to our attention some of the problems studied in this abstract.

## References

[ACGK11] H. Alt, S. Cabello, P. Giannopoulos, and C. Knauer. On some connection problems in straight-line segment arrangements. In *Abstracts of the 27th European Workshop on Computational Geometry (EuroCG 11)*, pages 27–30, 2011.

[BFC04] M. A. Bender and M. Farach-Colton. The level ancestor problem simplified. *Theor. Comput. Sci.*, 321(1):5–12, 2004.

[BK09] S. Bereg and D. G. Kirkpatrick. Approximating barrier resilience in wireless sensor networks. In *Proc. 5th ALGOSENSORS*, volume 5804 of *LNCS*, pages 29–40. Springer, 2009.

[BLWZ05] H. Broersma, X. Li, G. Woeginger, and S. Zhang. Paths and cycles in colored graphs, p.299. *Australasian J. Combin.*, 31:299–312, 2005.

[CdVL10] S. Cabello, É Colin de Verdière, and F. Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. In *Proc. 26th ACM SoCG*, pages 156–165, 2010.

[CEG$^+$94] B. Chazelle, H. Edelsbrunner, M. Grigni, L. J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12(1):54–68, 1994.

[CFL$^+$08] J. Chen, F. V. Fomin, Y. Liu, S. Lu, and Y. Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74:1188–1198, 2008.

[CLMS04] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discr. & Comp. Geometry*, 31(1):61–81, 2004.

[dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discr. & Comp. Geometry*, 14(3):261–286, 1995.

[dFOP91] H. de Fraysseix, P. Ossona de Mendez, and J. Pach. Representation of planar graphs by segments. *Intuitive Geometry*, 63:109–117, 1991.

[FGI10] M. R. Fellows, J. Guo, and I. Iyad. The parameterized complexity of some minimum label problems. *J. Comput. Syst. Sci.*, 76:727–740, 2010.

[GKV11] M. Gibson, G. Kanade, and K. Varadarajan. On isolating points using disks. Manuscript available at `http://arxiv.org/abs/1104.5043`, 2011.

[GSS89] L. J. Guibas, M. Sharir, and S. Sifrony. On the general motion-planning problem with two degrees of freedom. *Discr. & Comp. Geometry*, 4:491–521, 1989.

[Hås01] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[HMS07] R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *J. Comb. Optim.*, 14(4):437–453, 2007.

[HNZ91] I. B. Hartman, I. Newman, and R. Ziv. On grid intersection graphs. *Discrete Mathematics*, 87:41–52, 1991.

[HS99] J. Hershberger and S. Suri. An optimal algorithm for euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.

[HT84] D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2):338–355, 1984.

[KH07] S. Kloder and S. Hutchinson. Barrier coverage for variable bounded-range line-of-sight guards. In *Proc. ICRA*, pages 391–396. IEEE, 2007.

[KKMO07] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007.

[KLA07] S. Kumar, T.-H. Lai, and A. Arora. Barrier coverage with wireless sensors. *Wireless Networks*, 13(6):817–834, 2007.

[Mou90] David M. Mount. The number of shortest paths on the surface of a polyhedron. *SIAM J. Comput.*, 19(4):593–611, 1990.

[MT01] B. Mohar and C. Thomassen. *Graphs on surfaces.* Johns Hopkins Studies in the Mathematical Sciences. John Hopkins University Press, 2001.

[Tho90] C. Thomassen. Embeddings of graphs with no short noncontractible cycles. *J. of Comb. Theory, B*, 48(2):155–177, 1990.

[Tse11] K.-C. R. Tseng. Resilience of wireless sensor networks. Master's thesis, The University Of British Columbia (Vancouver), April 2011.

[Vaz01] V. V. Vazirani. *Approximation Algorithms.* Springer, 2001.