Most vital segment barriers

Irina Kostitsyna[1], Maarten Löffler[2], Valentin Polishchuk[3], and Frank Staals[2]

[1] Dept. of Mathematics and Computer Science, TU Eindhoven, the Netherlands
i.kostitsyna@tue.nl
[2] Dept. of Information and Computing Sciences, Utrecht University, the Netherlands
{m.loffler,f.staals}@uu.nl
[3] Communications and Transport Systems, ITN, Linköping University, Sweden
polishchuk@liu.se

Abstract. We study continuous analogues of "vitality" for discrete network flows/paths, and consider problems related to placing segment barriers that have highest impact on a flow/path in a polygonal domain. This extends the graph-theoretic notion of "most vital arcs" for flows/paths to geometric environments. We give hardness results and efficient algorithms for various versions of the problem, (almost) completely separating hard and polynomially-solvable cases.

Keywords: simple polygon, geodesic distance, flows and paths

1 Introduction

This paper addresses the following kind of questions:

> *Given a polygonal domain with an "entry" and an "exit", where should one place a given set of "barriers" so as to decrease the maximum entry-exit flow as much as possible ("flow" version), or to increase the length of the shortest entry-exit path as much as possible ("path" version)?*

Figure 1 illustrates these questions in their simplest form (placing a single barrier in a simple polygon). We call the solutions to the problems *most vital* segment barriers for the flow and the path resp. The name derives from the notion of *most vital arcs* in a network – those whose deletion decreases the flow or increases the length of the shortest path as much as possible. While the graph problems are well studied [1–4, 16, 18, 22, 27], to our knowledge, geometric versions of locating "most vital" facilities have not been explored. Throughout the paper, the segment barriers will be called simply barriers. When several segments are aligned to form a longer barrier, we call this longer segment a *super-barrier*. We focus only on segment barriers because already with segments there are a number of interesting problem versions, and in principle, any polygonal barrier may be created from sufficiently many segments; however, our results imply that the optimal blocking is always attained by gluing the barriers into super-barriers (no other configuration of segments is most vital).

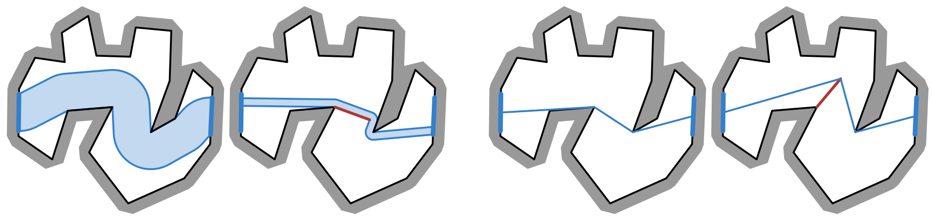

Fig. 1. A polygon in which a single barrier is placed to minimize the flow between two edges of the polygon (left) or lengthen the shortest path between two points (right).

Determining the most vital barriers is related to resilience and critical infrastructure protection, as it identifies the most vulnerable spots ("bottlenecks, weakest links") in the environment by quantifying how fragile or robust the flow/path is, how much it can be hurt, in the worst case, due to an adversarial act. It is thus an example of optimizing from an *adversarial* point of view: do as much harm as possible using available budget. In practice, the abstract "bad" and "good" may swap places, e.g., when the "good guys" build a defense wall, under constrained resources, to make the "evil" (epidemics, enemy, predator, flood) reach a treasure as late as possible (for the path version) or in a small amount (for the flow version). Our problem may also be viewed as a Stackelberg game (in networks/graphs parlance aka *interdiction problems* [8, 11, 13, 28, 30], extensively studied due to its relation to security) where the leader places the blockers and the follower computes the maximum flow or the shortest path around them.

Our paper also contributes to the plethora of work on uncertain environments [7, 17, 24]. Motion planning under uncertainty is important, e.g., in computing aircraft paths: locations of hazardous storm systems and other no-fly zones are not known precisely in advance, and it is of interest to understand how much the path or the whole traffic flow may be hurt, in the worst case, if new obstacles pop up (of course, there are many other ways to model weather uncertainty).

Finally, similar types of problems arise when barriers are installed for managing the queue to an airline check-in desk or controlling the flow of spectators to an event entrance.

Taxonomy. Since our input consists of the domain and the barriers, several problem versions may be defined:

H/h The domain may have an arbitrary number of holes (such versions will be denoted by H) or a constant number of holes (denoted by h)
B/b There may be arbitrarily many barriers (denoted B) or $O(1)$ barriers in the input (denoted b)
D/1 The barriers may have different lengths (denoted D) or all have the same length – w.l.o.g. unit (denoted 1)

Overall, for each of the two problems—flow blocking and path blocking—we have 8 versions (HBD, HB1, HbD, Hb1, hBD, hB1, hbD, hb1); e.g., flow-hBD is

Table 1. When the number of holes and barriers exceeds 1, the problem may become (weakly or strongly) NP-hard. This table shows which combinations of parameters lead to polynomial or hard problems. The results for $Hb1$, hbD and $hb1$ follow directly from the result for HbD.

	HBD	HB1	HbD	Hb1	hBD	hB1	hbD	hb1
Path	NP-hard	weakly NP-hard	poly	poly	weakly NP-hard	?	poly	poly
Flow	NP-hard	pseudo-poly	poly	poly	weakly NP-hard	poly	poly	poly

the problem of blocking the flow in a polygonal domain with $O(1)$ holes using arbitrarily many barriers of different lengths, etc. We allow barriers to intersect the holes. Depending on the nature of the barriers and the environment, in some of the envisioned applications these may be impractical (e.g., if a hole is pillar in the building, a barrier cannot run through it) while in others the assumptions are natural (e.g., if a hole is a pond near the entrance to an event). From the theoretical point of view, in most of our problems these assumptions are w.l.o.g. because in the optimal solution the barriers just touch the holes, not "wasting" their length inside a hole (one exception is HBD in which the solution may change if the barriers must avoid the holes).

Overview of the results. Section 3 describes our main technical contribution: a linear-time algorithm for the fundamental problem of finding *one* most vital barrier for the shortest s-t path in a *simple* polygon. The algorithm is based on observing that the barrier must be "rooted" at a vertex of the polygon. The main challenge is thus to trace the locations of the barrier's "free" endpoint (the one not touching the polygon boundary) through the overlay of shortest path maps from s and t. The overlay has quadratic complexity, so instead of building it, we show that only a linear number of the maps' cells can be intersected and work out an efficient way to go through all the cells. Furthermore, we prove that when placing multiple barriers they can be lined up into a single super-barrier; this reduces the problem to that of placing one barrier. In the remainder of the paper we consider polygons with holes. Section 4 shows hardness of the most general problems flow-HBD and path-HBD, i.e., blocking with multiple different-length barriers in polygons with (a large number of) holes. We also prove weak hardness of the versions with small number of holes (flow-hBD and path-hBD). Finally, we argue that path blocking is weakly hard if the barriers have the same length (path-HB1). Section 5 presents polynomial-time algorithms for path blocking with few barriers (path-HbD), implying that path-hbD, path-Hb1 and path-hb1 are also polynomial. The section then describes polynomial-time algorithms for the remaining versions of flow blocking. We first show that the problem is pseudopolynomial if the barriers have the same length (flow-HB1). We then prove that blocking with few barriers (flow-HbD) is strongly polynomial, implying that flow-hbD, flow-Hb1 and flow-hb1 are also polynomial. Finally, we show polynomiality of the version with constant number of holes (flow-hB1). Table 1 summarizes the hardness and polynomiality of our results.

2 Preliminaries

Let P be a polygonal domain with n vertices, and let the *source* S and the *sink* T be two given edges on the outer boundary of P (Fig. 2). A *flow* in P is a vector field $F : P \to \mathbb{R}^2$ with the following properties: $\operatorname{div} F(p) = 0\ \forall p \in P$ (there are no source/sinks inside the domain), $F(p) \cdot \mathbf{n}(p) = 0\ \forall p \in \partial P \setminus \{S \cup T\}$ where $\mathbf{n}(p)$ is the unit normal to the boundary of P at point p (the flow enters/exits P only through the source/sink), and $|F(p)| \leq 1\, \forall p \in P$ (the permeability of any point is 1, i.e., not more than a unit of flow can be pushed through any point—the flow respects the capacity constraint). Similarly to the discrete network flow, the *value* of a continuous flow F is the total flow coming in from the source ($\int_S F{\cdot}\mathbf{n}\ \mathrm{d}s$) – since in the interior of P the flow is divergence-free (flow conserves inside P), by the divergence theorem, the value is equal to the total flow out of the sink ($-\int_T F \cdot \mathbf{n}\ \mathrm{d}t$). A *cut* is a partition of P into 2 parts with S, T in different parts (analogous to a cut in a network); the *capacity* of the cut is the length of the boundary between the parts. Finally, the source and the sink split the outer boundary of P into two parts called the *bottom* $\mathcal{B}$ and the *top* $\mathcal{T}$, and the *critical graph* of the domain [10] is the complete graph on the domain's holes, $\mathcal{B}$ and $\mathcal{T}$, whose edge lengths equal to the distances between their endpoints (we assume that the edges are embedded to connect the closest points on the corresponding holes, $\mathcal{B}$ or $\mathcal{T}$). The celebrated Flow Decomposition and MaxFlow/MinCut theorems for network flows have continuous counterparts: (the support of) a flow decomposes into (thick) paths [25], and the maximum value of the S-T flow is equal to the capacity of the minimum cut [29]; moreover, the mincut is defined by the shortest $\mathcal{B}$-$\mathcal{T}$ path in the critical graph [21].

For shortest path blocking, the setup is a bit more elaborated. Let s be a point on the outer boundary of P, and let S^* be the edge containing s. We assume that s is actually an infinitesimally small gap s^-s^+ in the boundary of P (with s^- below s and s^+ above), and that the union of the barriers and the holes is not allowed to contain a path that starts on S^* below s^- and ends on

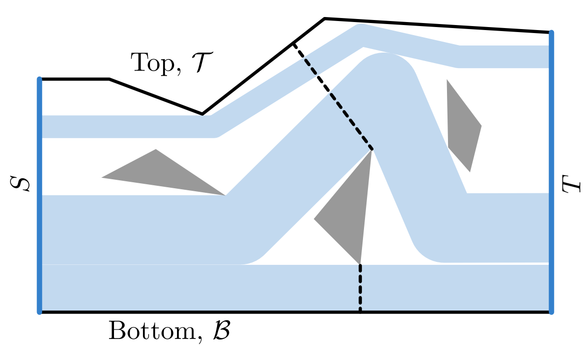

Fig. 2. Flow setup. An S-T flow decomposed into 3 thick paths (blue); two edges of the critical graph, defining a cut (dashed).

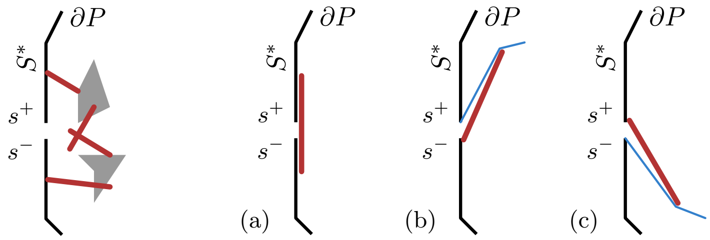

Fig. 3. Path setup; barriers are red and s-t path is blue. Surrounding s^-s^+ (left) is forbidden, even if no barrier touches the gap. Completely "shutting the door" s^-s^+ with one barrier (right (a)) is not allowed: if a barrier is at s, it must touch at most one of s^-, s^+ (right (b,c)).

S^* above s^+, completely cutting out s (Fig. 3).[4] W.l.o.g. we treat s^- and s^+ as vertices of P. Similarly, we are given a point t, modeled as a gap t^+t^- in another edge T^* on the outer boundary of P.

Let $\mathsf{SP}(p, q)$ denote a shortest path (a geodesic) between points p and q in P. Where it creates no confusion, we will identify a path with its length; in particular, for two points p, q, we will use pq to denote both the segment pq and its length. The *shortest path map* from s, denoted $\mathsf{SPM}(s)$, is the decomposition of P into cells such that shortest paths $\mathsf{SP}(s, p)$ from s to all points p within a cell visit the same sequence of vertices of P; the last vertex in this sequence is called the *root* of the cell and is denoted by $\mathsf{r}_s(p)$. The shortest path map from t ($\mathsf{SPM}(t)$) and the roots of its cells ($\mathsf{r}_t(p)$) are defined analogously. The maps have linear complexity and can be built in $O(n \log n)$ time (in $O(n)$ time if P is simple) [20]. Our algorithm for path blocking in a simple polygon uses:

Lemma 1. *[26, Lemma 1] Let p, q, and r be three points in a simple polygon P. The geodesic distance from p to a point $x \in \mathsf{SP}(q, r)$ is a convex function of x.*

Finally, let $E(u, v, p)$ denote the ellipse with foci u and v, going through the point p. It is well known that the sum of distances to the foci is constant along the ellipse; for the points outside (resp. inside) the ellipse, the sum is larger (resp. smaller) than $up + pv$. It is also well known that the tangent to the ellipse at p is perpendicular to the bisector of the angle upv (the light from u reaches v after reflecting from the ellipse at p).

[4] Other modeling choices could have been made; e.g, another way to avoid complete blockage could be to introduce a "protected zone" around s à la in works on *geographic mincut* [23]. Also a more generic view, outside our scope, could be to combine the flow and path problems into considering *minimum-cost* flows [25, 9] (the shortest path is the mincost flow of value 0) and explore how the barriers could influence both the capacity of the domain and the cost of the flow.

3 Linear-time algorithms for simple polygons

In this section P is a simple polygon. For a set $X \subset P$, let $\mathsf{SP}_X(p,q)$ denote the shortest path between points p,q in $P \setminus X$ (and the length of the path), i.e., the shortest p-q path avoiding X. We first consider finding the most vital unit barrier for the shortest path, i.e., finding the unit segment ab maximizing $\mathsf{SP}_{ab}(s,t)$. For the path blocking, we (re)define the bottom $\mathcal{B}$ and top $\mathcal{T}$ of P as the t^--s^- and s^+-t^+ parts of ∂P resp. (which mimics the flow setup, replacing the entrance S and exit T with s^-s^+ and t^-t^+). We will treat s^-, s^+, t^-, and t^+ as vertices of P. We then prove that a most vital barrier is placed at a vertex of P (Section 3.1). We focus on placing the barrier at (a vertex of) $\mathcal{B}$; placing at $\mathcal{T}$ is symmetric. In Section 3.2 we test whether it is possible for any unit barrier ab touching $\mathcal{B}$ to also touch $\mathcal{T}$ (while not lying on S^* or T^*): if this is possible, the barrier separates s from t completely and $\mathsf{SP}_{ab}(s,t) = \infty$. We test this by computing the Minkowski sum of $\mathcal{B}$ with a unit disk and intersecting the resulting shape with $\mathcal{T}$, taking special care around s and t (to disallow having $ab \subset S^*$). In Section 3.3 we then proceed to our main technical contribution: showing how to optimally place a barrier touching (a vertex of) $\mathcal{B}$ given that no such barrier can simultaneously touch $\mathcal{T}$. For this, we compute the shortest s-t path H around the Minkowski sum of $\mathcal{B}$ with the unit disk and argue that an optimal barrier will have one endpoint on (a vertex of) $\mathcal{B}$ and the other endpoint on H. Furthermore, we show that this path H intersects edges of the shortest path maps $\mathsf{SPM}(s)$ and $\mathsf{SPM}(t)$ only linearly many times. We subdivide H at these intersection points, and show that for each edge e of H we can then calculate the optimal placement of a point on e maximizing the sum of distances to s and t. This gives us a linear-time algorithm for finding a single most vital barrier. In Section 3.4 we then show that even if we have multiple barriers, it is best to glue the barriers together into a single super-barrier.

3.1 A most vital barrier is "rooted" at a vertex of P

We start by establishing the following lemma. It's complete proof can be found in the full version of this paper [15].

Lemma 2. *There exists a most vital barrier ab in which one endpoint, say b, lies on a vertex of P.*

Proof sketch. The main idea is to first show that there is a most vital barrier that touches ∂P, then that we can shift it to touch ∂P in an endpoint, and finally that we can shift it along ∂P until it's endpoint coincides with a vertex. □

3.2 Blocking the path from s to t completely

We now argue that we can check in linear time whether it is possible to completely block passage from s to t, by placing a barrier that connects $\mathcal{B}$ to $\mathcal{T}$ (without placing the barrier along S^* or T^*, which is forbidden by our model; see Section 2).

Observation 1. *Let u and v be two vertices of $\mathsf{SP}(s,t)$ in $\mathcal{B}$. The geodesic makes a right turn at u if and only if it makes a right turn at v. Let u' and v' be two vertices of $\mathsf{SP}(s,t)$ in $\mathcal{T}$. The geodesic makes a left turn at u' if and only if it makes a left turn at v'. Moreover, if $\mathsf{SP}(s,t)$ makes a right turn in u then it makes a left turn in u'.*

Assume w.l.o.g. that $\mathsf{SP}(s,t)$ makes a right turn at a vertex $u \in \mathcal{B}$. By Observation 1 it thus makes right turns at all vertices of $\mathsf{SP}(s,t) \cap \mathcal{B}$, and left turns at all vertices of $\mathsf{SP}(s,t) \cap \mathcal{T}$.

Observation 2. *If $\mathsf{SP}(s,t)$ makes a right turn at $u \in \mathcal{B}$, and we place a barrier ur at u, then $\mathsf{SP}_{ur}(s,t)$ makes a right turn at r.*

For every point p on $\mathcal{B}$, consider placing a barrier pq of length at most one, with one endpoint on p. The possible placements D_p of the other endpoint, q, form a subset of the unit disk centered at p. Let $\mathcal{D} = \bigcup_{p\in\mathcal{B}} D_p$ denote the union of all these regions (see Fig. 4).

Observation 3. *There is a barrier that separates s from t if and only if s and t are in different components of $P \setminus \mathcal{D}$.*

We now observe that $\mathcal{D}$ is essentially the Minkowski sum of $\mathcal{B}$ with a unit disk D. More specifically, let $A \oplus B = \{a + b \mid a \in A \wedge b \in B\}$ denote the Minkowski sum of A and B, let $S^*_{\mathcal{B}} = S^* \cap \mathcal{B}$ denote the part of S^* in $\mathcal{B}$, let $S^*_{\mathcal{T}}$, $T^*_{\mathcal{B}}$, and $T^*_{\mathcal{T}}$ be defined analogously, and let $\mathcal{B}' = \mathcal{B} \setminus (S^*_{\mathcal{B}} \cup T^*_{\mathcal{B}})$.

Lemma 3. *We have that $\mathcal{D} = \mathcal{D}' \cup X_S \cup X_T$, where $\mathcal{D}' = \mathcal{B}' \oplus D$, $X_A = (A^*_{\mathcal{B}} \oplus D) \setminus A^*_{\mathcal{T}}$, and D is the unit disk centered at the origin. Moreover, $\mathcal{D}$ can be computed in $O(n)$ time.*

Proof. The equality follows directly from the definition of $\mathcal{D}$ and the Minkowski sum. It then also follows $\mathcal{D}$ has linear complexity. So we focus on computing $\mathcal{D}$. To this end we separately compute $\mathcal{D}'$, X_S, and X_T, and take their union. More specifically, we construct the Voronoi diagram of $\mathcal{B}'$ using the algorithm of Chin, Snoeyink, and Wang [6], and use it to compute $\mathcal{B}' \oplus D$ [14]. Both of these steps can be done in linear time. Since S^*, T^*, and D have constant complexity, we can compute X_S and X_T in constant time. The resulting sets still have constant complexity, so unioning them with $\mathcal{B}' \oplus D$ takes linear time. □

Lemma 4. *We can test if s and t lie in the same component C of $P \setminus \mathcal{D}$, and compute C if it exists, in $O(n)$ time.*

Proof. Using Lemma 3 we compute $\mathcal{D}$ in linear time. If s or t lies inside $\mathcal{D}$, which we can test in linear time, then C does not exist. Otherwise, by definition of X_S and X_T, s and t must lie on the boundary of $\mathcal{D}$. We then extract the curve σ connecting s to t along the boundary of $\mathcal{D}$, and test if σ intersects the top of the polygon $\mathcal{T}$. If (and only if) σ and $\mathcal{T}$ do not intersect, their concatenation delineates a single component C' of $P \setminus \mathcal{D}$. Since C' contains both s and t we have $C = C'$. So, all that is left is to test if σ and $\mathcal{T}$ intersect. This can be done in linear time by explicitly constructing C' and testing if it is simple [5]. □

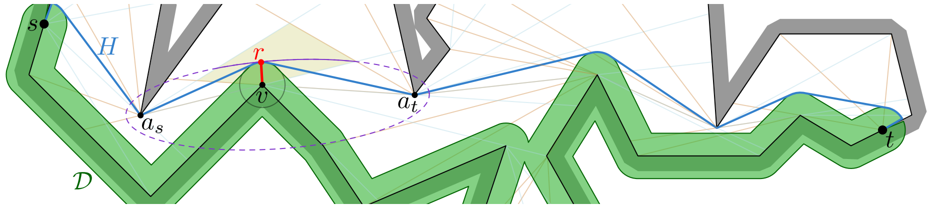

Fig. 4. Our algorithm constructs the region $\mathcal{D}$ describing possible placements of a barrier incident to $\mathcal{B}$, and the shortest path H around $\mathcal{D}$. An optimal barrier incident to $\mathcal{B}$ has one endpoint on H.

Theorem 4. *Given a simple polygon P with n vertices and two points s and t on the boundary of P, we can test whether there exists a placement of a unit length barrier that disconnects s from t in $O(n)$ time.*

3.3 Maximizing the length from s to t with a single barrier

In the remainder of the section we assume that we cannot place a barrier on (a vertex of) $\mathcal{B}$ that completely separates s from t. Fix a distance d, and consider all points $p \in P$ such that $\mathsf{SP}(s,p) + \mathsf{SP}(p,t) = d$. Let C_d denote this set of points, and define $C_{\leq d} = \bigcup_{d' \leq d} C_{d'}$.

Observe that an optimal barrier will have one of its endpoints on the boundary of $\mathcal{D}$. Let $H = \mathsf{SP}_{\mathcal{D}}(s,t)$ be the shortest path from s to t avoiding $\mathcal{D}$. We will actually show that there is an optimal barrier V^* whose endpoint a lies on H, and that H has low complexity. This then gives us an efficient algorithm to compute an optimal barrier. To show that a lies on H we use that if V^* realizes detour d^* (i.e., $\mathsf{SP}_{V^*}(s,t) = d^*$), the endpoint a also lies on C_{d^*}. First, we prove some properties of C_{d^*} towards this end.

Observation 5. *Let Δ_s be a cell in $\mathsf{SPM}(s)$ with root a_s, and Δ_t be a cell in $\mathsf{SPM}(t)$ with root a_t. We have that $C_d \cap \Delta_s \cap \Delta_t$ consists of a constant number of intervals along the boundary of the ellipse with foci a_s and a_t.*

Proof. A point $p \in C_d$ satisfies $\mathsf{SP}(s,p) + \mathsf{SP}(p,t) = d$. For $p \in \Delta_s \cap \Delta_t$ we thus have $\mathsf{SP}(s,a_s) + \|a_s p\| + \|p a_t\| + \mathsf{SP}(a_t,t) = d$. Since d, $\mathsf{SP}(s,a_s)$, and $\mathsf{SP}(a_t,t)$ are constant, this equation describes an ellipse with foci a_s and a_t. Since Δ_s and Δ_t have constant complexity the lemma follows. □

Lemma 5. *$C_{\leq d}$ is a geodesically convex set (it contains shortest paths between its points).*

Proof. Let p and q be two points on C_d, and assume, by contradiction, that there is a point r on $\mathsf{SP}(p,q)$ outside of $C_{\leq d}$. By Lemma 1 the geodesic distance from s to $\mathsf{SP}(p,q)$ is a convex function. Similarly, the distance from t to $\mathsf{SP}(p,q)$ is convex. It then follows that the function $f(x) = \mathsf{SP}(s,x) + \mathsf{SP}(x,t)$, for x on $\mathsf{SP}(p,q)$ is also convex, and thus has its local maxima at p and/or q. Contradiction. □

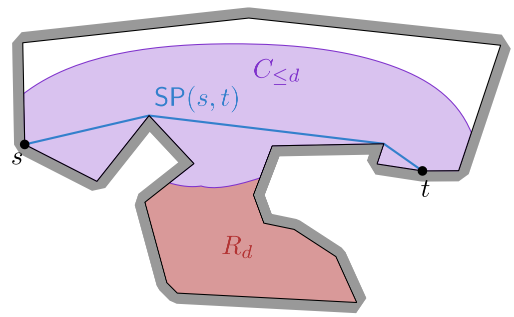

Fig. 5. A sketch of the regions $C_{\leq d}$ (purple) and R_d. Observe that R_d cannot contain any vertices of $\mathcal{T}$, otherwise $\mathcal{T}$ would have to pierce $\mathsf{SP}(s,t)$ and thus $C_{\leq d}$.

Lemma 6. *If there is an optimal barrier ua incident to a vertex u of $\mathcal{B}$, then the ray ρ from u through a intersects H.*

Proof. The ray ρ splits P into two subpolygons P_1 and P_2. Since $\mathsf{SP}_{ua}(s,t)$ makes a right bend at a (Observation 2 and our assumption that $\mathsf{SP}(s,t)$ makes a right turn at u) it intersects both subpolygons P_1 and P_2. It is easy to show that therefore s and t must be in different subpolygons (otherwise the geodesic crosses ρ a second time, and we could shortcut the path along ρ). Since H connects s to t it must thus also intersect ρ. □

Next, we define the region R_d "below" $C_{\leq d}$. More formally, let R' be the region enclosed by $\mathcal{B}$ and $\mathsf{SP}(s,t)$, let $d \geq \mathsf{SP}(s,t)$, and let $R_d = R' \setminus C_{\leq d}$. See Fig. 5. We then argue that it is separated from the top part of our polygon $\mathcal{T}$, which allows us to prove that there is an optimal barrier with an endpoint on H.

Observation 6. *Region R_d contains no vertices of $\mathcal{T}$.*

Proof. Assume, by contradiction that there is a vertex of $\mathcal{T}$ in R_d. Observe that this disconnects $C_{\leq d}$. However, since $C_{\leq d}$ is geodesically convex (Lemma 5) and non-empty it is a connected set. Contradiction. □

Lemma 7. *If there is an optimal barrier ua where u is a vertex of $\mathcal{B}$, then there is an optimal barrier ur where r is a point on $D_u \cap H$ (recall that D_u is the unit disk centered at u).*

Proof. Assume, by contradiction, that there is no optimal barrier incident to u that has its other endpoint on H. Consider the ray from u in the direction of a. By Lemma 6, the ray hits H in a point r' (Fig. 6(a)). Because a lies on C_{d^*} and $C_{\leq d^*}$ is geodesically convex (Lemma 5), r' lies outside $C_{\leq d^*}$. Let $H[p,q] = \mathsf{SP}_{\mathcal{D}}(p,q)$ be the maximal (open ended) subpath of H that contains r' and lies outside of $C_{\leq d^*}$. We then distinguish two cases, depending on whether or not $H[p,q]$ intersects (touches) $\mathcal{D}$:

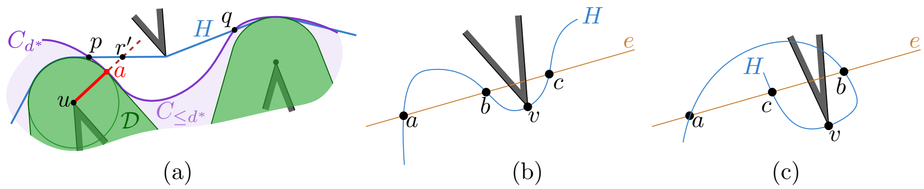

Fig. 6. (a) Illustration of Lemma 7. (b) and (c) The two cases in the proof of Lemma 8.

$H[p, q]$ **does not intersect (touch)** $\mathcal{D}$**.** It follows that $H[p, q]$ is a geodesic in P as well, i.e. $H[p, q] = \mathsf{SP}(p, q)$. Since $p, q \in C_{\leq d^*}$, and $C_{\leq d^*}$ is geodesically convex (Lemma 5) we then have that $H[p, q] \subseteq C_{\leq d^*}$. Contradiction.

$H[p, q]$ **intersects** $\mathcal{D}$ **in a point** z**.** Let $v \in B$ be a point such that $z \in D_v$. We distinguish two subcases, depending on whether z lies in the region R_{d^*}.

- $z \in R_{d^*}$**.** In this case z lies "below" $C_{\leq d^*}$. From $z \in H$ it follows that $H[p, q] \subset R_{d^*}$. However, as C_{d^*} is geodesically convex, this must mean that $H[p, q]$ has a vertex w in R_{d^*} at which it makes a left turn. This implies that w is a vertex of $\mathcal{T}$. By Observation 6 there are no vertices of $\mathcal{T}$ in R_{d^*}. Contradiction.
- $z \notin R_{d^*}$**.** Observe that vz is a valid candidate barrier. Since $z \notin C_{\leq d^*}$, the point z actually lies above (i.e. to the left of) $\mathsf{SP}(s, t)$, and thus $\mathsf{SP}_{vz}(s, t)$ makes a right turn at z. Using that $z \notin C_{\leq d^*}$ it follows that $\mathsf{SP}_{vz}(s, t) > d^*$. This contradicts that d^* is the maximal detour we can achieve.

Since all cases end in a contradiction this concludes the proof. □

We now know there exists an optimal barrier with an endpoint on H. Next, we focus on the complexity of H.

Observation 7. *Let b and c be two points on H, such that H makes a left turn in between b and c (i.e. the subcurve $H[b, c]$ of H between b and c intersects the half-plane right of the supporting line of bc). Then $H[b, c]$ contains a vertex of $\mathcal{T}$.*

Lemma 8. *The curve H intersects an edge e of $SPM(z)$, with $z \in \{s, t\}$, at most twice. Hence, H intersects $\mathsf{SPM}(z)$ at most $O(n)$ times.*

Proof. If e is a polygon edge, then H cannot intersect e at all, so consider the case when e is interior to P. Assume, by contradiction, that H intersects e at least three times, in points a, b, and c, in that order along H (Fig. 6(b) and (c)).

If the intersections a, b, and c, are also consecutive along e, then H makes both a left and right turn in between a and c. It is easy to see that since H can bend to the left only at vertices of $\mathcal{T}$ (Observation 7), the region (or one of the two regions) enclosed by H and ac must contain a polygon vertex. Since both e and $H[a, c]$ lie inside P, this means that P has a hole. Contradiction.

If the intersections are not consecutive, (say a, c, b), then again there is a region enclosed by $H[a, c]$ and ab, containing a polygon vertex. Since both $H[b, c]$ and cb lie inside P, this vertex must lie on a hole. Contradiction. □

Algorithm. We compute intersections of H with the shortest path maps $\mathsf{SPM}(s)$ and $\mathsf{SPM}(t)$, and subdivide H at each intersection point. By Lemma 8, the resulting curve H' still has only linear complexity. Consider the edges of H' in which H' follows the boundary of D_v, for the vertices v of $\mathcal{B}$. By Lemma 7 for some $v \in \mathcal{B}$ there is an optimal barrier that has one endpoint on such an edge of H' and the other at v. Since H' has only $O(n)$ edges we simply try each edge e of H'. For all points $r \in e$, the geodesics $\mathsf{SP}(s,r)$ and $\mathsf{SP}(t,r)$ have the same combinatorial structure, i.e., the roots $a_s = \mathsf{r}_s(r), a_t = \mathsf{r}_t(r)$ stay the same. It follows that we have a constant-size subproblem in which we can compute an optimal barrier in constant time. Specifically, we compute the smallest ellipse E with foci a_s and a_t that contains e and goes through the point r in which E and $e = D_v$ have a common tangent (if such a point exists). See Fig. 4. For that point r, we then also know the length of the shortest path $\mathsf{SP}_{vr}(s,t) = \mathsf{SP}(s,r) + \mathsf{SP}(r,t)$, assuming that we place the barrier vr. We then report the point r that maximizes this length over all edges of H'.

Constructing the connected component P' of $P \setminus \mathcal{D}$ that contains s and t takes linear time (Lemma 4). This component P' is a simple splinegon, in which we can compute the shortest path H connecting s to t in $O(n)$ time [19]. Computing $\mathsf{SPM}(s)$ and $\mathsf{SPM}(t)$ also requires linear time [12]. We can then walk along H, keeping track of the cells of $\mathsf{SPM}(s)$ and $\mathsf{SPM}(t)$ containing the current point on H. Computing the ellipse, the point p on the current edge e, and the length of the geodesic takes constant time. It follows that we can compute an optimal barrier incident to $\mathcal{B}$ in linear time. We use the same procedure to compute an optimal barrier incident to $\mathcal{T}$. We thus obtain the following result.

Theorem 8. *Given a simple polygon P with n vertices and two points s and t on ∂P, we can compute a unit length barrier that maximizes the length of the shortest path between s and t in $O(n)$ time.*

3.4 Using multiple vital barriers

We prove a structural property that even when we are given many barriers, there always exists an optimal solution in which they glued into a single super-barrier. This implies that our linear-time algorithm from the previous section can still be used to solve the problem.

Clearly, any solution distributes the barriers over some (unknown) number of super-barriers. First observe that, similarly to Section 3.1, any super-barrier must have a vertex at a vertex of P. We only need to argue that it is suboptimal to have more than one such super-barrier. Let a_1b_1 and a_2b_2 be two segments inside P, and let $m_1 \in a_1b_1, m_2 \in a_2b_2$ be two points that divide the segments in the same proportion, that is $\boldsymbol{m_1} = \gamma\boldsymbol{a_1} + (1-\gamma)\boldsymbol{b_1}, \boldsymbol{m_2} = \gamma\boldsymbol{a_2} + (1-\gamma)\boldsymbol{b_2}$ for some $\gamma \in [0,1]$. Then we may argue, similar to Lemma 1, that $f(\gamma) = \mathsf{SP}(m_1, m_2)$ is convex for $\gamma \in [0,1]$. The result then follows from multiple application of the triangle inequality, using that $\mathsf{SP}(m_1,m_2) \leq \gamma\mathsf{SP}(a_1,a_2) + (1-\gamma)\mathsf{SP}(b_1,b_2)$. The complete argument can be found in the full version of this paper [15].

Theorem 9. *Given a simple polygon P with n vertices, two points s and t on ∂P, and k unit-length barriers, the optimal placement of the barriers which maximizes the length of the shortest path between s and t consists of a single super-barrier.*

3.5 Most vital barriers for the flow

In simple polygons the critical graph has only two vertices – $\mathcal{B}$ and $\mathcal{T}$ (which, for the flow blocking, are the T-S and S-T parts of ∂P; refer to Fig. 2). Flow blocking thus boils down to finding the shortest $\mathcal{B}$-$\mathcal{T}$ connection (then all the barriers will be placed along the connecting segment) – a problem that was solved in linear time in [21].

4 Hardness results

In the remainder of the paper P is a polygonal domain with holes (as defined in Section 2). We show that when there are many barriers, it is hard to decide whether full blockage can be achieved, by reduction from Partition which reduces to deciding whether equal-width channels between S and T can be blocked (the reduction for path-HB1 is more involved, as it is not based on deciding full blockage); the details are in full version [15]. We summarize our results in the following two theorems.

Theorem 10. *Flow-HBD and path-HBD are NP-hard.*

Theorem 11. *Path-HB1, Flow-HBD, and path-hBD are weakly NP-hard.*

Membership in NP for our problems is open, since verifying solutions involve summing square roots.

5 Polynomial-time algorithms

For path blocking, we show that $O(1)$ barriers have only a polynomial number of “combinatorially different” placements and for each placement the different-homotopy path lengths are given by fixed functions of the barriers' locations. Flow blocking is reduced to shortening the $\mathcal{B}$-$\mathcal{T}$ path in the critical graph. The details are in the full version. We summarize our results in the following theorems.

Theorem 12. *Path-HbD, and hence path-hbD, path-Hb1 and path-hb1, are polynomial.*

Theorem 13. *Flow-HB1 can be solved in pseudopolynomial time.*

Theorem 14. *Flow-HbD, and hence flow-hbD, flow-Hb1 and flow-hb1, are polynomial.*

Theorem 15. *Flow-hB1 is polynomial.*

6 Conclusion

We introduced geometric versions of the graph-theoretic most vital arcs problem. We presented efficient solutions for simple polygons, and gave hardness results and algorithms for various versions of the problem. The most intriguing open problem is the hardness of path-hB1 (path blocking with few holes, our only unresolved version); we conjecture that it is polynomial, as still only a constant-number of super-barriers may be needed. Another interesting question is whether the flow and the path blocking have fundamentally different complexities: we proved that the complexities are the same for all versions except HB1 – for path-HB1 we showed weak hardness but lack a pseudopolynomial-time algorithm, while for flow-HB1 we have a a pseudopolynomial-time algorithm but no (weak) hardness proof. More generally, various other setups may be considered. For instance, one may be given a budget on the total length of the barriers– the problem then is how to split the budget between the barriers and where to locate them. For minimizing the maximum flow this version is easy: just place the barriers along the shortest $\mathcal{B}$-$\mathcal{T}$ path in the critical graph of the domain. For maximizing the shortest path in a simple polygon the solution is trivial: make a single barrier of the full length (and use our algorithm to find the optimal barrier location). Blocking shortest paths in polygons with holes in this setting is an open problem.

Acknowledgements M.L. and F.S. were partially supported by the Netherlands Organisation for Scientific Research (NWO) through project no 614.001.504 and 612.001.651, respectively.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications. Prentice hall (1993)
2. Alderson, D.L., Brown, G.G., Carlyle, W.M., Cox Jr, L.A.: Sometimes there is no most-vital arc: assessing and improving the operational resilience of systems. Tech. rep., Naval Postgraduate School Monterey CA (2013)
3. Ball, M.O., Golden, B.L., Vohra, R.V.: Finding the most vital arcs in a network. Operations Research Letters **8**(2), 73 – 76 (1989)
4. Bazgan, C., Nichterlein, A., Niedermeier, R.: A refined complexity analysis of finding the most vital edges for undirected shortest paths. In: International Conference on Algorithms and Complexity. pp. 47–60. Springer (2015)
5. Chazelle, B.: Triangulating a simple polygon in linear time. Discrete Compututational Geometry **6**(5), 485–524 (1991)
6. Chin, F., Snoeyink, J., Wang, C.A.: Finding the medial axis of a simple polygon in linear time. Discrete Comput. Geom. **21**(3), 405–420 (1999)
7. Citovsky, G., Mayer, T., Mitchell, J.S.B.: TSP With Locational Uncertainty: The Adversarial Model. In: 33rd International Symposium on Computational Geometry. Leibniz International Proceedings in Informatics (LIPIcs), vol. 77, pp. 32:1–32:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2017)
8. Collado, R.A., Papp, D.: Network interdiction–models, applications, unexplored directions. Rutcor Res Rep, RRR4, Rutgers University, New Brunswick, NJ (2012)

9. Eriksson-Bique, S., Polishchuk, V., Sysikaski, M.: Optimal geometric flows via dual programs. In: Proceedings of the thirtieth Annual Symposium on Computational Geometry. p. 100. ACM (2014)
10. Gewali, L., Meng, A., Mitchell, J.S.B., Ntafos, S.: Path planning in $0/1/\infty$ weighted regions with applications. ORSA J. Comput. **2**(3), 253–272 (1990)
11. Golden, B.: A problem in network interdiction. Naval Research Logistics (NRL) **25**(4), 711–713 (1978)
12. Guibas, L.J., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.E.: Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica **2**, 209–233 (1987)
13. Guo, Q., An, B., Tran-Thanh, L.: Playing repeated network interdiction games with semi-bandit feedback. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 3682–3690. IJCAI'17, AAAI Press (2017)
14. Kim, D.S.: Polygon offsetting using a Voronoi diagram and two stacks. Computer-Aided Design **30**(14), 1069 – 1076 (1998)
15. Kostitsyna, I., Löffler, M., Staals, F., Polishchuk, V.: Most vital segment barriers. CoRR **abs/1905.01185** (2019)
16. Lin, K.C., Chern, M.S.: Finding the most vital arc in the shortest path problem with fuzzy arc lengths. In: Multiple Criteria Decision Making, pp. 159–168. Springer (1994)
17. Löffler, M.: Existence and computation of tours through imprecise points. International Journal of Computational Geometry and Applications **21**(1), 1–24 (2011)
18. Lubore, S.H., Ratliff, H., Sicilia, G.: Determining the most vital link in a flow network. Naval Research Logistics (NRL) **18**(4), 497–502 (1971)
19. Melissaratos, E.A., Souvaine, D.L.: Shortest paths help solve geometric optimization problems in planar regions. SIAM Journal on Computing **21**(4), 601–638 (1992)
20. Mitchell, J.S.B.: Geometric shortest paths and network optimization. In: Sack, J.R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 633–701. Elsevier (2000)
21. Mitchell, J.S.: On maximum flows in polyhedral domains. Journal of Computer and System Sciences **40**(1), 88 – 123 (1990)
22. Nardelli, E., Proietti, G., Widmayer, P.: A faster computation of the most vital edge of a shortest path. Information Processing Letters **79**(2), 81–85 (2001)
23. Neumayer, S., Efrat, A., Modiano, E.: Geographic max-flow and min-cut under a circular disk failure model. Computer Networks **77**, 117–127 (2015)
24. Papadimitriou, C., Yannakakis, M.: Shortest paths without a map. In: Proc. 16th ICALP. pp. 610–620 (1989)
25. Polishchuk, V., Mitchell, J.S.: Thick non-crossing paths and minimum-cost flows in polygonal domains. In: Proceedings of the Twenty-third Annual Symposium on Computational Geometry. pp. 56–65. SCG '07, ACM (2007)
26. Pollack, R., Sharir, M., Rote, G.: Computing the geodesic center of a simple polygon. Discrete & Computational Geometry **4**(6), 611–626 (Dec 1989)
27. Ratliff, H.D., Sicilia, G.T., Lubore, S.: Finding the n most vital links in flow networks. Management Science **21**(5), 531–539 (1975)
28. Smith, J.C., Prince, M., Geunes, J.: Modern network interdiction problems and algorithms. In: Handbook of combinatorial optimization, pp. 1949–1987. Springer (2013)
29. Strang, G.: Maximal flow through a domain. Math. Program. **26**, 123–143 (1983)
30. Zhang, P., Fan, N.: Analysis of budget for interdiction on multicommodity network flows. J. of Global Optimization **67**(3), 495–525 (Mar 2017)