Decontaminating planar regions by sweeping with barrier curves

Borislav Karaivanov*, Minko Markov†, Jack Snoeyink‡, Tzvetalin S. Vassilev§

"If seven maids with seven mops
Swept it for half a year.
Do you suppose," the Walrus said,
"That they could get it clear?"
"I doubt it," said the Carpenter,
And shed a bitter tear.

Abstract

We consider the problem of decontaminating (cleaning) the interior of a planar shape by sweeping it with barrier curves. The contaminant is assumed to instantly travel any path not blocked by a barrier. We show that any decontamination sweep can be converted to one that uses only line segment barriers without increasing length. We define the *sweepwidth* of a region as the minimum over all decontamination sweeps of the maximum over time of barrier length used, and determine sweepwidth for some simple classes of orthogonal polygons. However, we also show that computing sweepwidth in general, even for orthogonal polygons, is $\mathcal{NP}$-hard.

1 Introduction

We consider a problem of decontaminating or cleaning a planar region (which we will consider a polygon, possibly with holes) by sweeping it with moving barriers, under the assumption that the contaminant spreads instantly whenever it is not blocked by barriers. This is a natural extension of the cops and robbers game on graphs where the goal is to minimize the overall number of cops used – minimizing the *node search number* first introduced by Kirousis and Papadimitriou [6]. The notion of *graph search* was introduced by Torrence Parsons in the 1970's [11]. For a detailed overview of different graph searches, see [3].

It is well known that graph searching is $\mathcal{NP}$-hard [9] in general. In fact, graph searching is $\mathcal{NP}$-complete even on planar graphs of maximum degree three [10], which are the direct discrete analogues of triangulated planar shapes. An optimal graph search can always be achieved without recontamination [8]; this is needed to establish $\mathcal{NP}$-completeness, rather than $\mathcal{NP}$-hardness, for graph searching, but is also interesting in its own right.

Many graph search problems have been generalized to polygon search problems. Often the aim is to reach the robber, spot the robber, or view the entire polygon [2, 7, 13]. Variants of the decontamination problem where both the contamination and the decontamination occur at the same time in an initially clean/empty polygon have been considered in the literature [1, 12]. We need a few definitions for our specific variant.

Definitions Suppose that we are given a compact, path-connected region S that is initially contaminated or dirty. (Think of S as a polygon, possibly with holes.) Define a dynamic set of moving barrier points in S, $b\colon [0,1] \to 2^S$, as a function from the unit interval to subsets of S. At time t the *barrier points* of $b(t)$ become clean, but once the barriers move, whenever there exists a path from a clean point p to a dirty point q that does not intersect a barrier point, point p is immediately recontaminated. Thus, *S is decontaminated* if and only if, for every continuous curve $\sigma\colon [0,1] \to S$ and continuous non-decreasing function $\tau\colon [0,1] \to [0,1]$ with $\tau(0) = 0$ and $\tau(1) = 1$, there exists some $t \in [0,1]$ such that $\sigma(t)$ is a barrier point at time $\tau(t)$.

We can clean any region S by sweeping with barrier curves that stretch from boundary to boundary. Using curves allows us to measure the length of the barriers employed, so let us restrict our attention to barriers that are piecewise continuous curves. That is, $b\colon [0,1]^2 \to S$ is a piecewise continuous function $b(s,t)$ in both curve parameter s and time t, and for any t, $b(\cdot,t)$ is 1-measurable. The *bottleneck length of a sweep* is the supremum over time on the sum of the lengths of all curves in the sweep. We look for the minimum bottleneck length of a decontamination sweep for S, which we call the *sweepwidth* of S and denote $\mathtt{sw}(S)$. For instance, an $a \times b$ rectangle has sweepwidth $\min(a,b)$, as shown in Corollary 6.

Alternatively, a decontamination sweep can be viewed as a collection of $2D$ surfaces $T \subseteq S \times [0,1]$ that *separates* $S \times [0,1]$ in the sense that any continuous time-monotone curve in $S \times [0,1]$ intersects T. In this view, bottleneck length is the supremum over $t \in [0,1]$ of the length of the intersection surface T with the plane at time t.

The problem of sweeping a simple polygon with a single barrier curve is essentially the problem of elastic ringwidth, which was solved in quadratic time by Yap [4, 14]. In contrast, our problem, which allows an arbitrary number of moving barriers, is $\mathcal{NP}$-hard by reduction from Partition. We leave as an open question whether every region S has a progressive decontamination of bottleneck length $\mathtt{sw}(S)$: whether there is always a way

*101 Saluda Pointe Dr., Lexington, SC, email: borislav.i.karaivanov@gmail.com

†Faculty of Mathematics and Informatics, Sofia University, Sofia, Bulgaria, email: minko.m.markov@fmi.uni-sofia.bg

‡Department of Computer Science, University of North Carolina at Chapel Hill, email: snoeyink@cs.unc.edu

§Department of Computer Science and Mathematics, Nipissing University, North Bay, Ontario, e-mail: tzvetalv@nipissingu.ca

to clean S and never have a point recontaminated. We conjecture that there is, but suggest that a proof may not be easy.

2 Canonical sweep

Our definitions of sweep are intentionally very general, but already have simple consequences:

Observation 1 *Time-reversing any decontamination sweep gives a decontamination sweep with the same bottleneck length.*

Lemma 1 *Let A and B be two compact, path-connected planar regions, such that $A \subseteq B$. Then, $\texttt{sw}(A) \leq \texttt{sw}(B)$.*

Proof. Take any sweep that cleans B. The intersection of its barriers with A cannot be longer. It also cleans A because any curve $\sigma\colon [0,1] \to A$ is a curve in B, and whenever $\sigma(t)$ is a barrier point at time $\tau(t)$, then it is a barrier in A. □

Corollary 2 *For any convex set A, $\texttt{sw}(A)$ cannot exceed the width of A, which is the minimum distance between two supporting lines.*

Proof. The infinite strip between two parallel lines at distance w can be swept with a length w barrier segment, so Lemma 1 says that $\texttt{sw}(A) \leq w$. □

In the full version, we expect to show that this is tight; that for convex regions $\texttt{sw}(A)$ equals the width, which would then simplify several of our lower bound arguments. (The proof of the result currently depends on an exhaustive enumeration of the ways to sweep triangles with up to two segments, which we don't have space or time for here.) This and other results that we do include, depend on the following theorem whose proof we sketch: we can focus our attention on sweeps that have a *canonical form*—the sweep at all times consists of one- and two-segment curves that start and end on the region border.

We claim that any decontamination sweep can be made canonical without increasing its length. The proof is by modifying a sweep in several steps. First, we eliminate extraneous barrier points—barrier points that are not on the boundary between dirty and cleaned portions of the region (dotted in Figure 1). We anchor each of the remaining barrier curves at its points along the region border, or at a highest point if there are no border points. We then replace barrier curves by shortest paths between anchor points. Usually anchor points move continuously; when they jump, we show how to interpolate so that the entire region is still swept.

Theorem 3 *From any decontamination sweep, we can create a decontamination sweep in canonical form without increasing length at any time.*

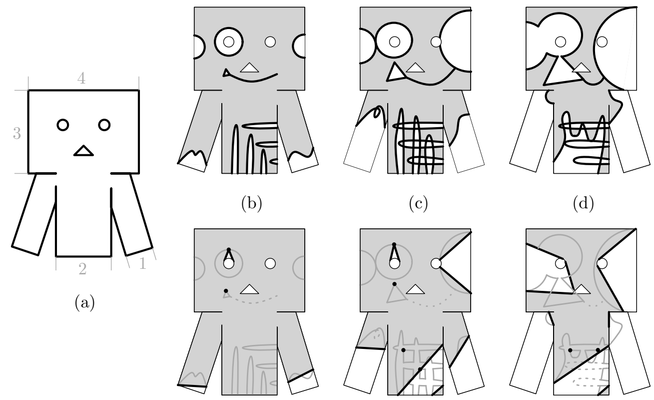

Figure 1: A "Danbo" polygon (a) with sweepwidth 4: sweep head to neck, each arm, then down torso. For the proof of Theorem 3, (b)–(d) show three steps of a decontamination sweep and their canonical forms.

Proof. Suppose that we have a decontamination sweep. At each time, each point on the boundary between the dirty and cleaned points is necessarily a barrier point; delete all barrier points that are not also boundary points, such as the "smile" in Figure 1.

The remaining barrier points can be grouped into curves that bound cleaned portions of the region. If multiple barrier curves meet at some point q, they are considered to be joined so that the dirty portion in the neighborhood of q is a single component. As Figure 2 suggests, this is like resolving a monkey saddle into simple saddles.

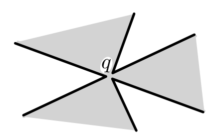

Figure 2: q's barriers made to bound a single dirty component

Barrier points that remain on the region border are considered *primary anchor* points. Any barrier/boundary curve that does not touch the region border is given a *top anchor* at its highest point; take the rightmost highest to break ties. Black dots in Figure 1(bcd) mark top anchors.

Any remaining barrier/boundary curve goes from anchor to anchor in the region. The next step replaces each such curve by the shortest path that is in the same homotopy class in the region—the shortest path that winds around holes in the region in the same way [5]. Shortest paths that contract to a point (that are *null homotopic*) are dropped, as are portions of shortest paths that follow the region border; several examples can be seen in Figure 1. Remaining points where the shortest paths touch the region border are made *secondary anchors.*

What remains after this step is a collection of line segments that cross the region, joining pairs of anchor points, at least one of which must be on the region border. Obviously, length cannot increase by taking shortest paths or dropping curve portions.

So far we have described how to make a single time step canonical. We must now connect these steps into a canonical decontamination sweep. The key is to observe the motion of anchors—when they move continuously, then the sweep by shortest paths is continuous. We sketch how how to interpolate when anchors do not move continuously, but are created by barrier curves hitting the region border, destroyed by barrier curves leaving the border, or jump because the highest point on a barrier curve changes.

Let me sketch the idea for the first of these: When a barrier curve hits a region border, one or more new anchor points are created; we think of them as being created in sequence, say ccw from some reference anchor point, r. (By outlawing wild curves, we will assume that when a barrier curve hits a border it does so in a finite number of connected portions, each of length greater than some $\epsilon > 0$.) We sweep the canonical curve in phases: first introducing an anchor point a on the curve, and joining it to r in a way that is a retract of the barrier curve. Then we move a to r maintaining the canonical curve as a shortest path. If there are more anchor points of the barrier curve connected to r, then we fix an anchor at r, but continue to move our anchor a to replace barriers with boundary, maintaining a shortest path for the portion of the barrier curve that is not fixed to the boundary.

The general idea for handling any discrete anchor change event is the same: determine the location of the event on the shortest path barrier curve and find the primary anchors to either side. Temporarily restore the original barriers between those anchors. Top anchors can be moved continuously from old to new position by sliding along these barriers, maintaining shortest paths. Anchors can be deleted by interpolating from the current shortest path to the shortest path with the anchor removed. And a new anchor can be added by moving it from a current anchor position, again maintaining shortest paths.

Multiple anchors may change simultaneously, but we can process them as a series of simple changes, much like a monkey saddle is decomposed into a sequence of simple saddles.

Finally, there are cases in which many barrier points reach the region border simultaneously—e.g., when sweeping a disk with a circular barrier growing from the center, no anchors would have been created. We can, at the final step, start at the highest border point and sweep by sliding two anchors away along the region border, maintaining shortest path between them. They will stop by meeting and joining other anchors or themselves. □

A simple corollary and a lemma establish lower bounds on sweepwidth for many specific regions that follow.

Corollary 4 *A point p at distance x from the region border needs segments at least $2x$ to sweep it.*

Proof. Any curve of at most two segments that reaches p with both ends on the border has length at least $2x$. □

Two immediate conclusions from Corollaries 2 and 4:

Corollary 5 *For a circle C, sweepwidth $\mathtt{sw}(C)$ equals the diameter of C.*

Corollary 6 *An $a \times b$ rectangle, R, with side lengths $a \leq b$ has $\mathtt{sw}(R) = a$.*

3 Sweepwidth for some orthogonal polygons

An *L-polygon* is the union of two rectangles, with dimensions $a \times w$ and $w \times b$ that share a vertex with the $a \times b$ rectangle that is their intersection, and and long 'arms,' say $w \geq a + b$. Key points are labeled in Figure 3.

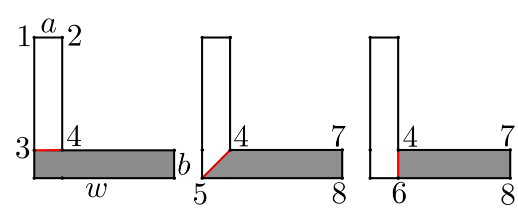

Figure 3: Optimal sweep of an L-polygon.

Lemma 7 *For an L-polygon with rectangles $a \times w$ and $w \times b$, where $w \geq a + b$, $sw(L) = \sqrt{a^2 + b^2}$.*

Proof. If we sweep, at any time, the two corridors of the L-polygon simultaneously, we incur length of at least $a + b$. Thus, one of the corridors must be swept first, and the other is swept afterwards. By Observation 1 we may assume, without loss of generality, that the sweep starts in position 1–2 at the left corridor of width a. If a sweep proceeds without reaching position 3–4, then it either allows recontamination, i.e. the corridor has to be swept again, or leaves a barrier of length at least a. Since the remaining part of the polygon includes a rectangle of dimensions $w \times b$, by Corollary 6 we incur length of at least $a + b$ if we leave the barrier in the left corridor. Thus, we may assume that we have reached position 3–4 in the sweep of the left corridor, so corner 4 is reached by the sweep. Consider now point 5. It can be cleaned as shown in Figure 3 at a cost of $\sqrt{a^2 + b^2}$. Any segments not containing point 4, while cleaning point 5 would have length of at least b in order to reach the barrier 3–4 (otherwise point 5 is recontaminated). In addition, any two chain segment connecting point 5 to point 4 or to the barrier 3–4 has length greater than the segment 4–5 by the triangle inequality. □

A *T-polygon* is the union of two rectangles, with dimensions $w \times a$ and $b \times w$, again with sufficiently long arms, say $w \geq 2a + b$, glued as in Figure 4; the short side of the $b \times w$ rectangle is glued to the middle of the long side of the $w \times a$ rectangle, and their interiors are disjoint.

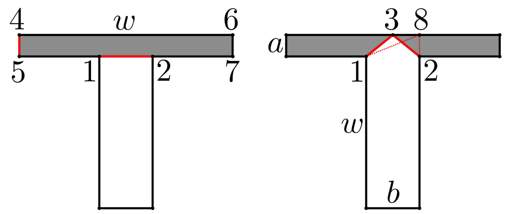

Figure 4: Optimal sweep of a T-polygon.

Lemma 8 *The sweepwidth of a T-polygon equals* $\sqrt{4a^2 + b^2}$ *when* $a \leq 2b/3$ *and* $a + b$ *otherwise.*

Proof. As in the proof of the previous Lemma 7, we observe that the vertical corridor of the T-polygon and either end of the horizontal corridor being swept at the same time immediately incurs sweepwidth of at least $a + b$. This is indeed optimal in some cases, as shown in Figure 4 on the left: the optimal sweep cleans the vertical corridor first, stopping in position 1–2. Then, the horizontal corridor is cleaned, say from left (4–5) to right (6–7). There is another possibility for the sweep to continue after the vertical corridor is cleaned, though. The barrier 1–2 can be split into two segments and the split point moved up until it reaches the opposite side of the horizontal corridor. The shortest way to achieve this is to move the split point along the perpendicular bisector of 1–2 until it reaches point 3. Since the lengths of the segments 1–3 and 2–3 are equal to $\sqrt{a^2 + (b/2)^2}$, the bottleneck length of this sweep is twice that, $\sqrt{4a^2 + b^2}$. Solving

$$\sqrt{4a^2 + b^2} \leq a + b$$

yields $a \leq 2b/3$. Note that moving the barrier 1–2 to position 1–8, as in the proof of Lemma 7, incurs greater length since segment 2–8 has to be made a barrier segment, too. $\square$

A *comb polygon*, shortly *comb*, is the union of $k+1$ axis-aligned rectangles: a $w \times d$ rectangle S called *the shaft* and k rectangles of size $q \times p_i$, denoted $T_1, \ldots, T_k$ and called *the teeth*, that lie along a common side, separated by at least b from each other and by d from the ends of the shaft. The dimensions have the following names and relations: q is *the length of a tooth*, $p_i < q$ is the *width of tooth* T_i, b is the minimum separation between teeth, $w \geq 2d + (k-1)b + \sum_{1 \leq i \leq k} p_i$ is the *length of the comb*, and $d < b$ is the *thickness of the shaft*. Teeth $T_1, \ldots, T_k$ are "glued" to the same long side of the shaft and lie in the opposite half-plane as the shaft. Figure 5 illustrates the structure of a comb polygon with these dimensions.

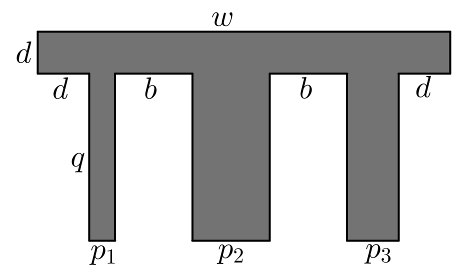

Figure 5: A three tooth comb. The shaft is horizontal and the teeth are vertical.

Lemma 9 *Let* $t = \max\{p_1, \ldots, p_k\}$. *If* $d > 2t/3$, *then the sweepwidth of the comb polygon is equal to* $d + t$.

Proof. The claim follows from the proof of Lemma 8. A decontamination sweep with bottleneck length $d + t$ that cleans the polygon starts on the left vertical side of the shaft, sweeps it with a vertical segment, and stops at the upper left corner of each tooth. The tooth is then cleaned by an upward sweep of a horizontal segment; once the upward sweep reaches the shaft, the left-to-right sweep of the shaft can continue. $\square$

Note that if $d \leq 2t/3$, we can start the sweep in the widest tooth and split it into the shaft as in the proof of Lemma 8. However, now the sweep progresses independently in the two parts of the comb to the left and to the right of the widest tooth. Whether we can finish the sweep within a length of $\sqrt{4d^2 + t^2}$ depends on what the dimensions of the other teeth are relative to t and where exactly they are placed relative to the end(s) of the two parts of the comb. Regardless of this, the ideas developed in this section help us modify the comb polygons in a way that significantly raises the complexity of their decontamination.

4 Reduction from Partition

Partition asks whether a multiset of n given positive integers $\{a_1, a_2, \ldots, a_n\}$ can be equi-partitioned into two multisets whose sums are equal. Note that if the sum is odd, the answer is negative.

We reduce Partition to sweep decontamination by constructing the gadget of Figure 6. Let $A = 1 + \sum_i a_i$ be one greater than the sum of all elements. Construct a rectilinear polygon consisting of a long *corridor* of size $A \times (n+3)(A+1)$ that is one unit below a series of $n+2$ square *rooms* of size $A \times A$, numbered $0 \ldots n+1$, and each one unit apart from its neighbors. Rooms 0 and $n+1$ are joined to the corridor by *doors* of length $1 \times (A-1)/2$; room $1 \leq i \leq n$ by a *door* of length $1 \times a_i$.

We ask whether this polygon can be cleaned by a sweep of length less than $2A$.

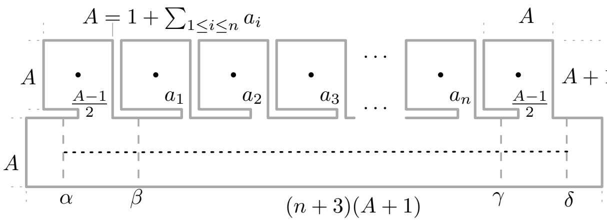

Figure 6: Our gadget for reducing Partition to sweep decontamination consists of $n+2$ square *rooms* joined by *doors* to a long *corridor*. The proof references the marked centers of rooms and corridor, and four line segment barriers, α–δ, drawn dashed.

Theorem 10 *An instance of* Partition *has an equi-partition if and only if the polygon of Figure 6 has a decontamination sweep of length less than* $2A$.

Proof. Suppose the Partition instance has an equi-partition. That is, there exist sets $S \subset [1..n]$ and $\overline{S}$ with $\sum_{i\in S} a_i = (A-1)/2$. Clean as follows: For each $i \in S$, sweep room i downward and leave a barrier at the door, then do the same for room 0. This sweep has maximum length $A + (A-1)/2$ and leaves barriers in doors of total length $A-1$. Next, sweep the corridor from left to right with a barrier segment of length A, giving sweep length of $2A-1$. As the barrier segment passes each door, it erases any door barrier that is present and creates any door barrier that is absent. Since it begins by erasing the barrier at door 0 (sweeping from dashed line α to β in Figure 6), the sweep length while sweeping doors 1 through n cannot exceed $A + \sum_{1\le i\le n} a_i = 2A-1$. Creating the final door (sweeping from dashed line γ to δ in Figure 6) returns the sweep length to $A + (A-1)/2 + \sum_{i\in\overline{S}} a_i = 2A-1$.

Note that reversing time would sweep the corridor right to left, placing the initial barriers in the complement of the doors; e.g., starting the corridor sweep by erasing a barrier in door $n+1$ and ending by creating the barrier in door 0.

Now we want to show that if there exists a canonical decontamination sweep with length $< 2A$, then there must be an equi-partition. By Corollary 4, any decontamination sweep that simultaneously sweeps a room center and a point on the centerline of the corridor must have length at least $2A$. Likewise for sweeping two room centers or two points $\ge A$ apart in the middle of the corridor. Thus, any cleaning strategy will sweep some set of rooms, sweep the corridor from end to end (aborted corridor sweeps that are picked up or that sweep forth and back can be ignored since the corridor will be recontaminated and must be swept again), then sweep the remaining rooms. By reversing time if necessary, we may assume that the corridor is swept (net) left to right.

Since the length of all doors $\sum_{0\le i\le n+1} a_i = 2A-2$, the best that can be done is to partition them. Choosing neither or both of doors 0 and $n-1$ would mean that the corridor sweep has total length $A+(A-1)+(A-1)/2 \ge 2A$ at position β or γ. Thus, a sweep of length $< 2A$ must choose one of these doors, and indicate a partition of $\{a_1, a_2, \ldots, a_n\}$. $\square$

5 Open questions

The main open question is whether the sweepwidth of a region can always be achieved by a *progressive sweep* that avoids recontamination altogether. All our examples so far certainly can be cleaned by progressive sweeps.

It is relatively easy to convert any decontamination sweep to a progressive sweep. Consider the view of a sweep as a collection of surface patches in $S \times [0,1]$, and fill in the portion of S that has been cleaned at each time. In a decontamination sweep, for any point $p \in S$, there is a time t_p at which p is last cleaned. Make p a barrier at that time. Unfortunately, this transformation does not preserve length.

To see a specific illustration, consider sweeping a regular n-gon from right to left with a vertical line, but spin the n-gon rapidly during the sweep (halting temporarily as the sweep passes through the width so as not to exceed the sweepwidth.) The spin means that the sweep line has to reach the center before a point is finally cleaned for the last time. Then, as the line continues, a cleaned disk grows from the center to cover the polygon. In $S \times [0,1]$, the helical spiral surface of the spinning sweep segment becomes a cone in conversion to a progressive sweep, so the barrier length is the circumference rather than the length of the n-gon. This shows that it will not be easy to convert an arbitrary decontamination sweep into a progressive one.

There are many other natural variations: the cost of a decontamination sweep could be the integral of length over time, or it could have a penalty for the number of connected components. The sweepers and the contaminant could have maximum travel speeds; some of these have been considered in grids [12]. We expect many variants to be $\mathcal{NP}$-hard as well.

Acknowledgement

We thank the referees for their detailed reading and helpful comments, and regret that due to sabbatical move timing we have been able to incorporate most, but not all good suggestions into this version.

References

[1] Y. Altshuler, V. Yanovski, I. A. Wagner, and A. M. Bruckstein. Multi-agent cooperative cleaning of expanding domains. *I. J. Robotic Res.*, 30(8):1037–1071, 2011.

[2] B. K. Bhattacharya, T. Kameda, and J. Z. Zhang. Surveillance of a polygonal area by a mobile searcher from the boundary: Searchability testing. In *ICRA*, pages 2461–2466. IEEE, 2009.

[3] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.

[4] J. E. Goodman, J. Pach, and C. K. Yap. Mountain climbing, ladder moving, and the ring width of a polygon. *Amer. Math. Monthly*, 96:494–510, 1989.

[5] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.*, 4:63–98, 1994.

[6] L. M. Kirousis and C. H. Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(3):205–218, 1986.

[7] K. Klein and S. Suri. Catch me if you can: Pursuit and capture in polygonal environments with obstacles. In *Proc. 26th Conf. Artificial Intel.*, pages 2010–2016, 2012

[8] A. S. LaPaugh. Recontamination does not help to search a graph. *J. ACM*, 40(2):224–245, 1993.

[9] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. ACM*, 35(1):18–44, Jan. 1988.

[10] B. Monien and I. Sudborough. Min cut is $\mathcal{NP}$-complete for edge weighted trees. *Theor. Comput. Sci.*, 58(13):209 – 229, 1988.

[11] T. Parsons. Pursuit-evasion in a graph. In Y. Alavi and D. Lick, editors, *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. Springer, 1978.

[12] E. Regev, Y. Altshuler, and A. M. Bruckstein. The cooperative cleaners problem in stochastic dynamic environments. *CoRR*, abs/1201.6322, 2012.

[13] J. Urrutia. Art gallery and illumination problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. Elsevier, 1999.

[14] C.-K. Yap. How to move a chair through a door. *IEEE J. Rob. Auto.*, RA–3(3):172–181, June 1987.