Improved Approximation Bounds for the Minimum Constraint Removal Problem

Sayan Bandyapadhyay[1]
Department of Computer Science, University of Iowa, Iowa City, USA
sayan-bandyapadhyay@uiowa.edu

Neeraj Kumar
Department of Computer Science, University of California, Santa Barbara, USA
neeraj@cs.ucsb.edu

Subhash Suri
Department of Computer Science, University of California, Santa Barbara, USA
suri@cs.ucsb.edu

Kasturi Varadarajan
Department of Computer Science, University of Iowa, Iowa City, USA
kasturi-varadarajan@uiowa.edu

Abstract

In the minimum constraint removal problem, we are given a set of geometric objects as obstacles in the plane, and we want to find the minimum number of obstacles that must be removed to reach a target point t from the source point s by an *obstacle-free* path. The problem is known to be intractable, and (perhaps surprisingly) no sub-linear approximations are known even for simple obstacles such as rectangles and disks. The main result of our paper is a new approximation technique that gives $O(\sqrt{n})$-approximation for rectangles, disks as well as rectilinear polygons. The technique also gives $O(\sqrt{n})$-approximation for the *minimum color path* problem in graphs. We also present some inapproximability results for the geometric constraint removal problem.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Minimum Constraint Removal, Minimum Color Path, Barrier Resilience, Obstacle Removal, Obstacle Free Path, Approximation

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2018.2

Funding Research of Sayan Bandyapadhyay, Neeraj Kumar and Subhash Suri was supported in part by the NSF grant CCF-1525817. Research of Sayan Bandyapadhyay and Kasturi Varadarajan was supported in part by the NSF grant CCF-1615845.

1 Introduction

Given a set $\mathcal{S}$ of geometric objects as *obstacles* in the plane, a path is called *obstacle-free* if it does not intersect the interior of any obstacle. In the *minimum constraint removal* (MCR) problem, the goal is to remove a minimum-sized subset $\mathcal{S}' \subseteq \mathcal{S}$ such that the remaining set $\mathcal{S} \setminus \mathcal{S}'$ admits an obstacle-free path between a source point s and the target point t. The problem is known to be NP-complete even when the obstacles have very simple geometry such as rectangles or line segments. The MCR problem is also related to the *minimum color path* (MCP) problem, where the goal is to find a path in a graph using the minimum number

[1] The work was partially done when the author was visiting University of California, Santa Barbara.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018).
Editors: Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer; Article No. 2; pp. 2:1–2:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of colors. In the vertex-colored version of the problem, each vertex v of a graph $G = (V, E)$ is associated with a set of colors $\chi(v) \subseteq \mathcal{C} = \{1, 2, \ldots, |\mathcal{C}|\}$, and the goal is to find a path between two fixed vertices s and t (s-t path) that minimizes the total number of colors along the path. Similarly, in the edge-colored version, each edge of G has an associated set of colors, and the s-t path must minimize the total number of colors along the path.

The geometric constraint removal problem can be cast as a minimum color path problem by constructing a graph on the *arrangement* formed by the obstacles. The arrangement $\mathcal{A}(\mathcal{S})$ of $\mathcal{S}$ is a partition induced by the obstacles, whose *faces* are the two-dimensional connected regions, and whose edges are segments of the obstacle boundaries. We now define a planar graph $G_{\mathcal{A}}$ whose vertices are in one-to-one correspondence with the faces of the arrangement, and whose edges join two neighboring faces. By associating each obstacle with a unique color, we obtain a version of the minimum color path problem—an s–t path has exactly as many colors along it as the number of obstacles it crosses. However, it is worth pointing out that the number of vertices in $G_{\mathcal{A}}$ can be *quadratic* in the size of the geometric input: a set of n geometric obstacles, each with a constant number of boundary edges, can create an $\Omega(n^2)$ size arrangement.

The minimum color path problem is also NP-complete, and by a reduction from Set Cover it is also NP-hard to approximate to a factor better than $o(\log n)$ even if the graph is planar [8, 14, 20]. In [13], Hassin et al. solve a special case of the MCP problem where each edge has *exactly* one color. For that special case, they present an $O(\sqrt{|V|})$-approximation algorithm. However, for the general MCP problem, no sub-linear algorithm appears to be known to the best of our knowledge.

The geometric minimum constraint removal problem has been studied under different names across multiple research communities, including sensor networks and robotics. In the sensor networks, the problem is called *barrier resilience*, in which a collection of sensors are modeled for providing (overlapping) geometric coverage in the plane, and the network's resilience is measured by the minimum number of sensors whose *removal* creates a *sensor-avoiding* s–t path. The most common form of geometric obstacles considered in sensor network applications are *circular disks*. When all disks have the same (unit) radius, a 2-approximation is known due to Chan and Kirkpatrick [5], who build and improve upon the earlier work of Bereg and Kirkpatrick [3]. However, even for this simple case, the complexity of minimum constraint removal is an unsolved open problem [5]. In [5, 17], constant factor approximations are proposed for *restricted versions* of arbitrary radii disks. However, in general, when disks have arbitrary radii, no sub-linear approximation with provable guarantee is known. The problem has also been studied for other types of obstacles, mainly from the perspective of time complexity. The problem has been shown to be NP-complete for convex obstacles [11], for line segments [19], even in the bounded density case [2, 10], and for axis-parallel rectangles with aspect ratio close to one [17].

In robotics, the minimum constraint removal problem models the motion planning problem of multi-articulated robot [11, 14]. Suppose we have a physical environment constaining a disjoint set of impenetrable obstacles in the plane, and a robot with two degrees of freedom. Then the *configuration space* approach to motion planning *shrinks* the robot to a point while simultaneously *expanding* the obstacle by taking their Minkowski sum with the robot's geometry. The result is our minimum constraint removal problem: a set of two-dimensional intersecting obstacles that may have no feasible path for the robot, and so some obstacles need to be removed.

Finally, the problem has also been studied through the lenses of parameterized complexity [10, 17], and exact and heuristic algorithms [9, 14]. It is also loosely related to a shortest path problem in the plane [1, 15], where given a set of disjoint obstacles, the goal is to find an Euclidean shortest s-t path that intersects at most k obstacles.

1.1 Our Results

In this work, we make progress on both the graph and the geometric versions of MCP by obtaining improved approximation results. All these approximations are achieved in polynomial time. For the minimum constraint removal problem, we obtain the following results.

- We present an $O(\sqrt{n})$-approximation for rectilinear polygons, where n denotes the total number of vertices of the polygons. No sublinear approximation was known even for squares.
- We present an $O(\sqrt{n})$-approximation for disks, where n denotes the number of disks. For arbitrary disks, the only approximation results known are in the restricted cases, where either the crossing patterns of the paths are limited or the aspect ratio and the density are bounded.

Prior approximation algorithms for MCR proceed by establishing a good bound on the number of times an optimal path enters a removed obstacle. For the obstacles we consider, i.e., rectilinear polygons and disks, this bound is very large. Thus the previous approaches are inadequate to obtain the above mentioned results. Our main new idea is to use a filtering step, which removes a small number of obstacles that are potentially expensive in terms of the number of times an optimal path enters those obstacles.

The above results are based on an algorithmic framework, which uses the filtering step mentioned before. As a byproduct, the framework also gives an $O(\sqrt{|V|})$-approximation for MCP on vertex-colored graphs.

We also obtain a few hardness results for MCR which give a better understanding of the problem. We show that for rectilinear polygons, the problem is NP-hard to approximate within a factor better than 2. The same result holds even for convex polygons. We also prove the APX-hardness of the problem in a more restricted case, where the obstacles are axis-parallel rectangles.

The framework is described in Section 2. The application of the framework to the MCR problem is discussed in Section 3. Finally we describe the hardness results in Section 4. Throughout this paper, the proofs of lemmas and theorems marked with $(*)$ are given in the appendix due to space constraints.

2 An Algorithmic Framework

We begin our discussion by introducing a generic framework that yields a sublinear approximation for minimum color path problems on graphs. In the later sections, we apply this framework to achieve similar approximation bounds for the MCR problem. Roughly speaking, the framework comprises of two main steps. In the first step, a 'small' subset of the colors are removed from the instance based on some conditions. In the second stage, an approximation of the minimum color path is computed using a shortest path algorithm. We start with some basic definitions.

As an input to the framework, we assume that we are given a graph $G = (V, E)$, the source vertex s, the target vertex t, and a set of colors $\mathcal{C}$, such that each vertex $v \in V$ is assigned a subset $\chi(v) \subseteq \mathcal{C}$ of colors. We will refer to such a graph as a *colored graph* and denote it by $G = (V, E, \mathcal{C})$. we define the set of colors $\chi(\pi)$ used by π to be the union of the colors of vertices on this path. That is, $\chi(\pi) = \bigcup_{v \in \pi} \chi(v)$.

▶ **Definition 1.** Any path π in G is a *k-color path* if the number of colors used $|\chi(\pi)|$ is k.

An algorithm is called an *α-approximation* algorithm for computing a k-color path if it satisfies the following two conditions: (1) if there exists a k-color path π from s to t, it computes a path π^* such that $|\chi(\pi^*)| \leq \alpha k$, and (2) if there is no k-color path from s to t, it returns an arbitrary s-t path. The following is straightforward.

▶ **Lemma 2.** *If there exists an α-approximation algorithm to compute a k-color path from s to t then there also exists an α-approximation algorithm for computing a minimum color path from s to t.*

Proof. We try all possible values $k = 1, 2, \ldots, |\mathcal{C}|$ and let π_k be the path returned by the approximation algorithm for computing a k-color path for a given value of k. Let j be the value such that $\chi(\pi_j)$ has smallest cardinality over all $\chi(\pi_k)$. Now, let l be the number of colors used by a minimum color path, then $|\chi(\pi_l)|$ must be at most αl. Clearly, $|\chi(\pi_j)| \leq |\chi(\pi_l)| \leq \alpha l$ and therefore π_j is an α-approximation for computing a minimum color path. ◀

From Lemma 2 it follows that computing an approximation of a k-color path is sufficient, and therefore in the rest of our discussion, we work towards that goal. Next, we describe the details of our framework.

2.1 Approximation Framework

As an input to the framework, we assume that we are given a colored graph $G = (V, E, \mathcal{C})$, and an integer k. The key idea behind our approximation framework is to define a notion of *neighborhood* for the colors in $\mathcal{C}$, and 'discard' the colors that have dense neighborhoods.

▶ **Definition 3.** Let $\mathcal{P}$ be an arbitrary set of objects and β be a parameter. We define *neighborhood* $\mathcal{N} : \mathcal{C} \to 2^{\mathcal{P}}$ to be a mapping from $\mathcal{C}$ to subsets of $\mathcal{P}$ that satisfies the following properties.

1. ***(Bounded-Size Property)*** Sum of cardinalities of all neighborhoods $\sum_{C \in \mathcal{C}} |\mathcal{N}(C)|$ is $O(k\beta^2)$
2. ***(Bounded-Occurrence Property)*** If there exists a k-color path in G, then there also exists a k-color path π^* in G such that, for any color $C \in \mathcal{C}$, the number of times C appears on π^* is at most $O(|\mathcal{N}(C)|)$.

The set $\mathcal{P}$ in the above definition can be any set of objects. For example, in MCP problem $\mathcal{P}$ is the set of vertices of the graph. In the geometric MCR, $\mathcal{P}$ is a set of points in the plane. We now describe our approximation algorithm which we will refer to as APPROX-CORE.

Algorithm APPROX-CORE

1. Construct the neighborhood $\mathcal{N}(C)$ for each color $C \in \mathcal{C}$.
2. For all $C \in \mathcal{C}$, remove all occurrences of the color C from the graph G if $|\mathcal{N}(C)| \geq \beta$. Let G' be the modified graph after removing all such colors.
3. For every vertex v in G', assign an integer weight $|\chi(v)|$ on v.
4. Compute a minimum weight path π from s to t in G' using Dijkstra's Algorithm. Return π.

▶ **Lemma 4.** *Given the set $\mathcal{P}$ and a parameter β, the algorithm APPROX-CORE gives an $O(\beta)$-approximation for the k-color path in G.*

Proof. Assume that there exists a k-color path in G. Otherwise, the proof is trivial as the algorithm always returns a path. Let $\mathcal{C}_1$ be the set of colors removed during step 2 of the algorithm, and $\mathcal{C}_2$ be the set of colors in G' that appear on the path π returned by the algorithm. Then, the total number of colors in G that may appear on π is at most $|\mathcal{C}_1| + |\mathcal{C}_2|$.

First we compute a bound on the size of $\mathcal{C}_1$. Observe that the neighborhood of each color $C \in \mathcal{C}_1$ has size at least β. Therefore, we have:

$$\begin{aligned} \sum_{C \in \mathcal{C}_1} |\mathcal{N}(C)| &\leq \sum_{C \in \mathcal{C}} |\mathcal{N}(C)| \\ \implies |\mathcal{C}_1| \cdot \beta &\leq O(k\beta^2) && \text{By the bounded-size property of } \mathcal{N} \\ \implies |\mathcal{C}_1| &\leq ck\beta && \text{for some constant } c \end{aligned}$$

Next, we compute a bound on size of $\mathcal{C}_2$. Towards this end, observe that the neighborhood $\mathcal{N}(C)$ of every color C in G' has fewer than β colors. By the bounded-occurrence property of the neighborhood $\mathcal{N}$, there exists a k-color path π^* in G such that for each $C \in \mathcal{C}$, the number of times C appears on π^* is at most $O(|\mathcal{N}(C)|)$. Therefore, it follows that any color C in G' appears on π^* at most $O(\beta)$ times. In other words, there exists a path in G' that has weight at most $c'k\beta$ for another constant c'. Therefore the number of colors used by the minimum weight path π is at most $c'k\beta$.

Hence, the total number of colors in $\mathcal{C}$ that appear on π is at most $|\mathcal{C}_1| + |\mathcal{C}_2| = (c + c') \cdot k\beta$, which is an $O(\beta)$-approximation. ◀

We obtain the following theorem.

▶ **Theorem 5.** *Given a colored graph $G = (V, E, \mathcal{C})$, suppose a neighborhood $\mathcal{N}$ for G can be constructed in polynomial time that satisfies the bounded-size and bounded-occurrence property. Then there exists a polynomial time algorithm that achieves an $O(\beta)$-approximation for computing a k-color path in G.*

Therefore, in order to achieve an approximation for the k-color path, it just suffices to construct a neighborhood $\mathcal{N}$, that satisfies the bounded-size and bounded-occurrence properties. In the next section, we illustrate this construction for MCP on vertex-colored graphs.

2.2 Application to Minimum Color Path

In this section, we will apply the above framework to achieve $O(\sqrt{n})-$approximation for MCP on a vertex-colored graph $G = (V, E, \mathcal{C})$ with n vertices. Our goal is to simply compute a neighborhood $\mathcal{N}$ for a k-color path in G such that $\mathcal{N}$ has bounded-size $O(kn)$ and satisfies the bounded-occurrence property. Using Lemma 2 and $\beta = \sqrt{n}$ in Theorem 5, an $O(\sqrt{n})$-approximation for MCP follows.

We define neighborhood $\mathcal{N}(C)$ of each color C to be the set $\{v \in V \mid C \in \chi(v)$ and $|\chi(v)| \leq k\}$. The bounded-occurrence property is easily satisfied because a k-color path π_k will never visit vertices that contain more than k colors, and since π_k is simple, each occurrence of a color C on the path can be uniquely charged to a vertex in $\mathcal{N}(C)$. To see that the bounded-size property is satisfied, we note the following.

$$\sum_{C \in \mathcal{C}} |\mathcal{N}(C)| = \sum_{v \in V : |\chi(v)| \leq k} |\chi(v)| \leq O(kn).$$

▶ **Theorem 6.** *There exists an $O(\sqrt{n})$-approximation algorithm for MCP on vertex-colored graphs.*

Application to Minimum Label Path. As another example application for the framework, we consider a special case of MCP when each edge has exactly one color (called its label). This problem has been well studied [4,12,13] under the name *minimum label path.* Hassin et al. [13] gave an $O(\sqrt{n})$-approximation for this problem on general graphs. Using our framework and the following simple definition of neighborhood, we can achieve an $O(\sqrt{\frac{n}{OPT}})$-approximation if the number of edges in G is $O(n)$. Here OPT is the number of labels used by any minimum label s-t path.

For the sake of applying the framework, we transform the input edge-colored graph $G = (V, E, \mathcal{C})$ into a vertex-colored graph H by adding a vertex corresponding to each edge. The color corresponding to an old edge is moved to the new vertex. Now, for each new vertex v that has color C, we include both neighbors (old vertices) of v in H to the neighborhood of C. The bounded-occurrence property is straightforward. For the bound on size, observe that an old vertex v can be in at most $degree(v)$ neighborhoods, so sum of cardinality of all neighborhoods is at most $2|E|$. Since $|E| = O(n)$, the size of $\mathcal{N}$ is $O(n) = O(\frac{n}{k} \cdot k)$. With $\beta = \sqrt{n/k}$, Theorem 5 and Lemma 2 give an $O(\sqrt{\frac{n}{OPT}})$-approximation.

3 Application to Geometric Objects

In this section, we apply our approximation framework to achieve sublinear approximation for MCR when the obstacles are rectilinear polygons (Section 3.1) and disks of arbitrary radii (Section 3.2). Observe that if m is the number of cells of the input arrangement $\mathcal{A}$ of obstacles in $\mathcal{S}$, applying Theorem 6 on the graph obtained from $\mathcal{A}$ easily gives a $O(\sqrt{m})$-approximation. However, m can be $\Omega(n^2)$ and therefore this approach does not give us an $O(\sqrt{n})$-approximation. Here n is the number of vertices if obstacles are polygonal or the number of disks otherwise. By exploiting the geometry of obstacles, we show how to construct a colored graph $G = (V, E, \mathcal{C})$ and a sparse neighborhood $\mathcal{N}$ even when the underlying colored graph G can have $\Omega(n^2)$ complexity.

Recall that, there are two main steps for applying the framework. First we need to construct the colored graph G such that an s–t path in the plane that removes the minimum number of constraints, corresponds to a path in G that uses the minimum number of colors. Next, we need to construct the neighborhood $\mathcal{N}$ for colors in G such that it satisfies the bounded-size and bounded-occurrence properties. Note that for technical reasons, the graph G we construct for the geometric instances has colors assigned on edges. Indeed, for the sake of applying the framework, one can easily transform it into a vertex-colored graph by adding a vertex corresponding to each edge. We begin by revisiting some necessary background.

Any arrangement of obstacles in the plane can be partitioned into two distinct regions namely the obstacles, and *free space*, that is the region of the plane not occupied by obstacles. Without loss of generality, we assume that the points s and t lie in free space, as we must remove all the obstacles that are incident to either s or t in order to find an obstacle free s–t path. We say that a path π *crosses* an obstacle S if π intersects the interior of S. Note that, as s and t lie in free space, if π crosses S, π must intersect the boundary of S transversally.

Consider an optimal path π that removes the minimum number of obstacles. It is easy to see that π will cross an obstacle S if and only if S was removed from input. Therefore, removing an obstacle is equivalent to crossing it. In the following, we introduce the notion of a *k-crossing path.*

▶ **Definition 7.** A path π in the plane is called a *k-crossing path* if it crosses exactly k obstacles.

It is easy to see that if each obstacle is assigned a unique color and we assign color to a path whenever it enters an obstacle, then a k-crossing path π uses exactly k colors. Observe that although the space of k-crossing paths is infinite, we want to establish a one to one correspondence between the path in the plane that crosses minimum number of obstacles and a path in G that uses the minimum number of colors.

▶ **Definition 8.** Given a set $\mathcal{S}$ of obstacles in the plane, a one to one mapping $\mathcal{M} : \mathcal{S} \to \mathcal{C}$ from a set of obstacles $\mathcal{S}$ to a set of colors $\mathcal{C} = \{1, 2, \ldots, |\mathcal{C}|\}$, we say that a graph $G = (V,\ E,\ \mathcal{M}(\mathcal{S}))$ with two fixed vertices v_s, v_t is a *valid colored graph* for the input arrangement if the following conditions hold:

1. if there is a k-color v_s-v_t path in G, then there is also a j-crossing s-t path in the plane for some $j \leq k$, and
2. if there is a k-crossing s-t path in the plane, then there is also a j-color path from v_s to v_t in G for some $j \leq k$.

The first condition is typically established by fixing an *embedding* for the edges of G in the plane. From the above discussion and using Lemma 2 and Theorem 5 with $\beta = \sqrt{n}$, we have the following.

▶ **Lemma 9.** *Suppose we are given a valid colored graph $G = (V, E, \mathcal{C})$ for an arrangement of the set $\mathcal{S}$ of input obstacles in the plane, such that there is a k-color path in G. If we can construct the neighborhoods $\mathcal{N}(S)$ for all obstacles $S \in \mathcal{S}$ such that, the total size across all neighborhoods is $O(kn)$* (bounded-size)*, and there exists a k-color path π in G from v_s to v_t such that for any obstacle $S \in \mathcal{S}$, the corresponding color appears on π at most $O(|\mathcal{N}(S)|)$ times* (bounded-occurrence)*, then* APPROX-CORE *achieves an $O(\sqrt{n})$-approximation for* MCR*.*

3.1 An $O(\sqrt{n})$-approximation for Rectilinear Polygons

We begin by describing our construction of a valid colored graph $G = (V,\ E,\ \mathcal{M}(\mathcal{S}))$ for the input set of obstacles $\mathcal{S}$. Without loss of generality, we assume that the mapping $\mathcal{M}$ simply assigns a unique color $C_i \in \mathcal{C}$ to each obstacle $S_i \in \mathcal{S}$.

Graph Construction. Let V be the set of vertices of all obstacles in $\mathcal{S}$ (including s and t). Let v_s and v_t be the vertices corresponding to the points s and t, respectively. We build a complete graph over this vertex set by adding an edge (u, v) to E for every pair of vertices $u, v \in V$. We define a *rectilinear embedding* of an edge $e = (u, v)$ in the plane as follows.

> Without loss of generality, assume u lies below and to the left of v, and let x be the point where a horizontal ray from u and a vertical ray from v intersect. The rectilinear path $\pi_{uv} = \overline{ux} \to \overline{xv}$ is called the embedding of edge e. We assign e the colors corresponding to obstacles, whose boundaries are intersected by π_{uv} transversally.

Roughly speaking, we assign a color to an edge if it intersects both the interior and boundary of the corresponding obstacle, i.e., when the edge enters or exits the obstacle. It is easy to see that with the above construction, the first condition of Definition 8 is satisfied. For the second requirement, we make the following claim.

▶ **Lemma 10.** *If there exists a k-crossing s-t path π^* in the plane, then there exists a v_s-v_t path π in G that uses at most k colors.*

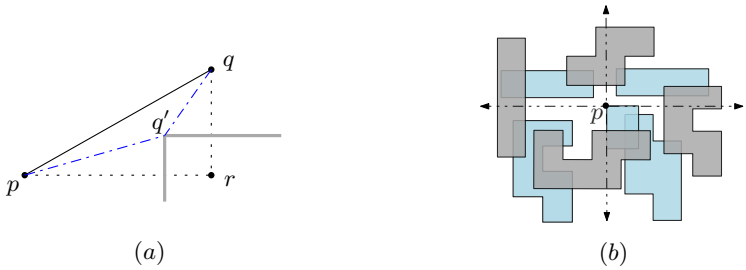

Figure 1 (a) Recursively modifying the path π to make it rectilinear. Observe that the segment $\overline{pq}$ cannot cross any obstacle edge. (b) Construction of the neighborhood for $k = 1$. p lies in the neighborhoods of all polygons shown in dark gray.

Proof. We prove this in three steps. First we construct a path π' from π^* such that π' consists of only straight line segments. Next we modify π' to obtain a rectilinear path $\pi_\perp$. Then we claim that $\pi_\perp$ is corresponding to a path π in G that uses at most k colors.

First, we remove all the k obstacles that are crossed by π^* which makes π^* lie entirely in free space connecting s and t. Since s and t are now connected in free space, there must exist a path of minimum Euclidean length that also lies in free space. Let this path be π'. It is easy to see that edges of π' are straight line segments, and lie in free space. Also π' bends only at obstacle vertices. We now recursively modify each segment $\overline{pq}$ of π to obtain a rectilinear path $\pi_\perp$. Let r be the point where a horizontal line through p intersects a vertical line through q. There are two cases. (See also Figure 1(a).)

1. If the triangle Δpqr contains one or more obstacle vertices, we find the vertex q' closest to the segment $\overline{pq}$, and replace $\overline{pq}$ with the segments pq' and $q'q$. It is easy to verify that π still lies in free space after this modification. We now repeat the process recursively on the segments pq' and $q'q$.
2. Otherwise, we simply replace $\overline{pq}$ by the rectilinear path $\pi_{pq} = pr \to rq$. Observe that since the obstacles are rectilinear and the segment $\overline{pq}$ lies in free space, no obstacle segment can intersect the triangle Δpqr, and therefore π_{pq} also lies in free space.

Observe that $\pi_\perp$ obtained using the above procedure lies in free space and crosses the boundaries of no more than k obstacles. Next, we retrieve an v_s-v_t path in G from this rectilinear path $\pi_\perp$. Note that, the way $\pi_\perp$ is constructed, at least one of the two endpoints of each of its segments is a vertex of a polygon and hence appears as a vertex in G. It follows that, the subpath between two such consecutive vertices along $\pi_\perp$ must consists of at most two rectilinear segments: one horizontal segment followed by a vertical segment. Hence, this subpath is the rectilinear embedding of the edge between the two vertices. We construct a path π by selecting the vertices along $\pi_\perp$ in order and connecting each consecutive pair of vertices by an edge. By definition, π is in G and uses no more than k colors. ◀

Construction of the Neighborhood. Now we construct the neighborhood $\mathcal{N}(S)$ for all obstacles $S \in \mathcal{S}$ that satisfies the bounded-size and bounded-occurrence properties. We choose the ground set $\mathcal{P}$ of elements in the neighborhood to be the set of vertices V of the obstacles in $\mathcal{S}$. Now, we define the *neighborhood* $\mathcal{N}(S)$ of an obstacle $S \in \mathcal{S}$ to be the subset of vertices from where one can reach a point on the boundary of S moving along a vertical or a horizontal segment and crossing no more than k obstacles. Roughly speaking, $\mathcal{N}(S)$ comprises of the vertices 'nearby' the boundary of S. We compute the set $\mathcal{N}(S)$ for every $S \in \mathcal{S}$ as follows.

For every vertex $v \in V$, draw four axis-aligned rays emanating at v one in each of the four directions. Next, find the first k distinct obstacles whose boundaries are intersected by each of these rays transversally as we move away from v. Let this set be X_v and therefore $|X_v| \leq 4k$. For each $S \in X_v$, include v to the neighborhood $\mathcal{N}(S)$. (See also Figure 1.(b))

The elements in the union of all neighborhoods is the set of vertices in the input and therefore has size $O(n)$. Moreover, since each element is present in at most $4k$ neighborhoods, $\sum_{S \in \mathcal{S}} |\mathcal{N}(S)|$ is $O(kn)$. Therefore the *bounded-size* property is easily satisfied. For the bounded-occurrence property, we prove the following lemma.

▶ **Lemma 11.** *Let π be any k-color v_s-v_t path in G and $S_i \in \mathcal{S}$ be an arbitrary obstacle. Then the color C_i corresponding to obstacle S_i appears on edges of π at most $O(|\mathcal{N}(S_i)|)$ times.*

Proof. We consider the set $E_i = \{e \mid C_i \in \chi(e)\}$ of edges that contain the color C_i and need to show that $|E_i| = O(|\mathcal{N}(S_i)|)$. Let $e = (p, q)$ be an arbitrary edge in E_i and let $\pi_{pq} = \overline{pr} \to \overline{rq}$ be its rectilinear embedding. From our construction, e has color C_i iff π_{pq} crossed S_i. Therefore, at least one of $\overline{pr}$ and $\overline{rq}$ must intersect the boundary of S_i transversally. This implies that at least one of p and q must be included in $\mathcal{N}(S_i)$ during our neighborhood construction. If $p \in \mathcal{N}(S_i)$, we charge this occurrence of color C_i to p, otherwise we charge it to q. Since a vertex $p \in \mathcal{N}(S_i)$ is adjacent to at most two edges in π, every element in $\mathcal{N}(S_i)$ is charged at most twice. Therefore C_i occurs on edges of π at most $2|\mathcal{N}(S_i)|$ times. ◀

Using Lemma 9, we obtain the following result.

▶ **Theorem 12.** *If all the obstacles in $\mathcal{S}$ are rectilinear polygons, then there exists an $O(\sqrt{n})$-approximation algorithm for the MCR problem.*

3.2 An $O(\sqrt{n})$-approximation for Arbitrary Disks

We will now consider the case when all the input obstacles are disks, of possibly different radii. The construction of the neighborhood for disks needs to be different, as the earlier arguments for rectilinear polygons relied heavily on obstacles having corners and therefore do not apply to disks. Recall that in order to apply our approximation framework, we first need to construct a valid colored graph $G = (V, E, \mathcal{C})$ (Lemma 9). Towards this end, we simply let G to be the graph $G_{\mathcal{A}}$ induced by the input arrangement $\mathcal{A}$: each cell of $G_{\mathcal{A}}$ contains a vertex and any pair of neighboring cells (vertices) are joined by an arc that does not intersect any other cell. Let v_s and v_t be the vertices in G corresponding to the cells that contain s and t, respectively. We assign a unique color to each disk. Additionally, we make G directed by replacing each edge by two directed edges. For each directed edge $e = (u, v)$, we assign to e the set of colors corresponding to all the disks D such that v lies in the interior of D and u does not lie in the interior of D. Roughly speaking, we assign colors when the edge *enters* into an obstacle.

Note that the way G is defined, it is a plane graph and we consider its natural embedding which is also planar. Since we assign colors when an edge of G enters an obstacle, it is easy to see that a k-color path π in G corresponds to a k-crossing path π' in the plane. For the other direction, given a k-crossing path π' we can easily construct a path π in G by simply concatenating the vertices corresponding to each arrangement cell intersected by π' in order. Thus, we have the following immediate observation.

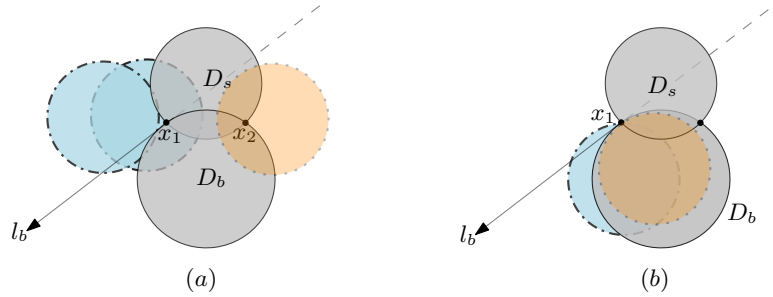

Figure 2 Disks shown in dash-dotted (shaded blue) are *critical* and included to $\mathcal{D}(D_s, D_b)$. Disks shown in dotted (shaded orange) are not critical with respect to the pair D_s, D_b.

▶ **Observation 13.** *G is a valid colored graph for the input disks.*

As each color is corresponding to a disk and vice versa, we will use the term disk instead of color in the context of applying the framework. In the following, we describe the neighborhood construction. The key idea behind our construction is to pick a subset of the $O(n^2)$ possible intersection points between pairs of disks in the input and use them for the ground set $\mathcal{P}$ of the neighborhood.

Computation of the Set $\mathcal{P}$ and the Neighborhood $\mathcal{N}$

Consider any two disks D_s, D_b such that their boundaries intersect transversally. Without loss of generality, assume that radius of D_s is smaller than D_b and their boundaries intersect at points x_1, x_2. Consider the arc $\widehat{x_1x_2}$ of ∂D_b (boundary of D_b) that lies inside D_s and assume x_1 lies before x_2 in clockwise traversal of this arc. Let l_b be the half line emanating at x_1 colinear to the tangent of D_b at x_1 and not intersecting D_s as shown in Figure 2. We now define the set of *critical* disks $\mathcal{D}(D_s, D_b)$ corresponding to a pair of intersecting disks D_s and D_b.

▶ **Definition 14** (Critical Disks)**.** Let D be a disk that intersects both D_s and D_b. We say that the disk D is *critical* with respect to the pair D_s, D_b if D intersects the half line l_b (see Figure 2).

We include the tuple (x_1, D_s, D_b) to the set of neighborhood candidates $\mathcal{P}$ if and only if the size of the critical set $\mathcal{D}(D_s, D_b)$ corresponding to the intersecting pair D_s, D_b is at most k. We are now ready to define the neighborhood $\mathcal{N}(D)$ of a disk D.

> For each disk D, add all the tuples of the form (x_1, D_s, D) to the neighborhood $\mathcal{N}(D)$. In addition, add a constant number of *phantom* points to the neighborhood $\mathcal{N}(D)$ of all disks D.

In the next few lemmas we establish the bounded-size and bounded-occurrence property of $\mathcal{N}$. We will need the notion of *exterior-disjointness*. Given a disk D and two disks D_1, D_2 that intersect D, we say that D_1 and D_2 are exterior-disjoint w.r.t D if the regions $D_1 \setminus D$ and $D_2 \setminus D$ are disjoint. A set $\mathcal{D}$ of disks is called exterior-disjoint w.r.t a disk D if any pair of disks in $\mathcal{D}$ is exterior-disjoint of D. We note the following.

▶ **Lemma 15.** *For a given disk D_s, let $\mathcal{D}$ be a set of disks such that the radius of any disk in $\mathcal{D}$ is at least the radius of D_s. If $\mathcal{D}$ is exterior-disjoint w.r.t D_s, then $|\mathcal{D}|$ is at most six.*

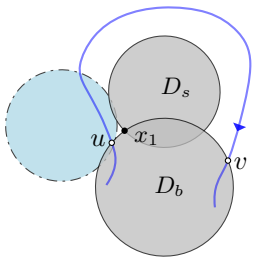

Figure 3 An entry of π_{uv} is charged to $x_1 \in \mathcal{N}(D_b)$.

Now we prove the bounded-size property of $\mathcal{N}$. We note that, if there are no three disks in the input whose boundaries intersect at a common point, then this proof follows readily from the bound of $O(kn)$ on the number of depth k points in an arrangement of disks [7, 16]. To see this, observe that for a tuple (x_1, D_s, D_b), that is added to $\mathcal{P}$, the intersection point of D_s and D_b must lie inside at most k other disks. Otherwise, the set $\mathcal{D}(D_s, D_b)$ contains more than k disks which is a contradiction. Moreover, since no more than two disks intersect at a point, each such intersection point adds at most two tuples. Thus the total size of all neighborhoods is at most two times the number of vertices in the arrangement of the input disks that lie inside at most k other input disks. From [7] and [16], it follows that the number of such vertices is $O(kn)$, and hence the bounded-size property follows. However, in presence of degeneracy, we might add as many as k tuples corresponding to one intersection point, and thus the bound does not follow immediately. Nevertheless, we prove the following lemma.

▶ **Lemma 16.** $(*)$ *The number of elements in the the set $\mathcal{P}$ is $O(kn)$.*

Since each element of $\mathcal{P}$ is included in a unique neighborhood, the bounded-size property of $\mathcal{N}$ follows from the above lemma. Next, we prove the bounded-occurrence property of $\mathcal{N}$. Observe that, it is sufficient to show the existence of a k-crossing s-t path π in the plane that enters into (and therefore crosses) any disk D at most $|\mathcal{N}(D)|$ times. This is because, an edge of G is assigned the color C corresponding to D only when it enters D, and thus every occurrence of C on the path π is corresponding to a crossing of D by π. Also existence of such a geometric k-crossing s-t path π implies the existence of a k-color v_s-v_t path in G where each color corresponding to a disk D appears at most $|\mathcal{N}(D)|$ times.

Now consider any k-crossing path π in the plane. By a similar argument as in Lemma 10, there also exists a k-crossing path π' such that the edges of this path are either straight line segments (tangents between a pair of disks) or parts of obstacle boundary (arcs). With this convention, one can now easily define the length of any such path as the sum of the lengths of its segments and arcs. Let π^* be the minimum length s-t path that crosses k disks. Then, we prove the following lemma.

▶ **Lemma 17.** *π^* crosses any disk D_b at most $O(|\mathcal{N}(D_b)|)$ times.*

Proofsketch. It suffices to prove that every entry (and therefore crossing) of the disk D_b by π^* can be charged to an element in $\mathcal{N}(D_b)$ so that every element is charged at most $O(1)$ times. Let $v_1, v_2, \ldots, v_l$ be the entry points on the disk D_b by π^*. We charge the first entry to a phantom point. Now consider the ith crossing for any $i > 1$. Let u_i be the point on π^* immediately before v_i where π^* last crossed D_b. Thus the subpath $\pi_{u_i v_i}$ of π^* between u_i and v_i lies in the exterior of D_b. Let $A_{u_i v_i}$ be the arc on the boundary of D_b with endpoints u_i and v_i such that the region $R_{u_i v_i}$ enclosed by the closed curve $\pi_{u_i v_i} \cup A_{u_i v_i}$ does not contain D_b.

Consider the set of arcs $\{A_{u_i v_i} \mid 2 \leq i \leq l\}$. First, assume that any two arcs in this set are disjoint. Since π^* is a minimum length k-crossing path, there must be at least one disk not crossed by π^* that intersects $R_{u_i v_i}$ as well as the disk D_b. Let D_s be the first such disk encountered while traversing the arc $A_{u_i v_i}$ clockwise along the boundary of D_b. There are two cases. If D_s is bigger than D_b, we charge the crossing to one of the phantom points in $\mathcal{N}(D_b)$. Otherwise, we assume D_s is smaller than D_b. Let x_1 be the first intersection point of ∂D_s and ∂D_b encountered while traversing the arc $A_{u_i v_i}$ clockwise along the boundary of D_b. Observe that π^* must cross all the disks in $\mathcal{D}(D_s, D_b)$ because otherwise it will contradict the choice of D_s.

Hence the size of $\mathcal{D}(D_s, D_b)$ is bounded by k, the number of disks that π^* can cross. This implies the tuple (x_1, D_s, D_b) must be included to the neighborhood $\mathcal{N}(D_b)$, and therefore we can charge this crossing to this tuple.

Because of the disjointness assumption of the arcs, the same tuple cannot be charged to another crossing.

Also it is easy to verify that the phantom points need not be charged more than a constant number of times, as the set of disks bigger than D_b for which we charge a phantom point must be exterior-disjoint of D_s and therefore can be at most six (Lemma 15). The case when the arcs are not disjoint, for any two arcs, one must contain the other. This is true, as the subpaths $\pi_{u_i v_i}$ as defined above cannot intersect each other. By exploiting this structure of the arcs one can prove the lemma in this case as well. ◀

Using Lemma 9, we obtain the following result.

▶ **Theorem 18.** *If all the obstacles in $\mathcal{S}$ are disks, then there exists an $O(\sqrt{n})$-approximation algorithm for the MCR problem.*

Using a different realization of the algorithmic approach described above, it appears possible to derive an approximation guarantee close to $O(\sqrt{n})$ for other obstacle types. We sketch this for the case where the obstacle set $\mathcal{S}$ is a set of triangles satisfying standard degeneracy assumptions. We define the *level* of a 2-dimensional cell σ in the arrangement of the triangles in $\mathcal{S}$ to be the minimum number of triangles whose removal results in an obstacle-free path from s to (any point in) σ. Thus, there is only one cell at level 0, and this is the cell containing s. We can show that the number of cells with level at most k is $O(kn\alpha(n))$, where $\alpha(\cdot)$ is the inverse Ackermann function. Furthermore, the number of arrangement edges bordering such cells is also $O(kn\alpha(n))$.

Suppose there is a k-crossing path from s to t, and we want to approximate it. For a triangle $T \in \mathcal{S}$, we include in its neighborhood $\mathcal{N}(T)$ any arrangement edge e such that (a) e is part of T's boundary, and (b) e borders a cell σ that is contained in T and has level at most k. It follows that the sum of the neighborhood sizes is $O(kn\alpha(n))$. We can also establish the bounded occurrence property, leading to an approximation guarantee of $O(\sqrt{n\alpha(n)})$. We defer the details to the journal version.

4 Hardness of Approximation

In this section, we describe the 2-inapproximability and the APX-hardness results for rectilinear polygons and axis-parallel rectangles, respectively. Due to space constraints, we will just mention the results and defer the details to the appendix.

2-inapproximability. We reduce an instance of Vertex Cover to an instance of MCR with rectilinear polygons. Since Vertex Cover is hard to approximate within a factor of 2 assuming the Unique Games conjecture [18], we get the following theorem.

▶ **Theorem 19.** *Minimum constraint removal with rectilinear polygons is hard to approximate within a factor of* 2 *assuming the Unique Games conjecture.*

The same construction can also be extended for convex polygons.

▶ **Corollary 20.** *Minimum constraint removal with convex polygons is hard to approximate within a factor of* 2 *assuming the Unique Games conjecture.*

APX-hardness for Axis Parallel Rectangles. We reduce a restricted version of vertex cover to our problem which is referred to as Special-3VC. Chan et al. [6] had introduced this version for the sake of proving APX-hardness of several geometric optimization problems. As Special-3VC is APX-hard we obtain the following theorem.

▶ **Theorem 21.** *Minimum constraint removal with rectangles is APX-hard.*

References

1. P. Agarwal, N. Kumar, S. Sintos, and S. Suri. Computing shortest paths in the plane with removable obstacles. In *16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018)*, pages 5:1–5:15, 2018.
2. H. Alt, S. Cabello, P. Giannopoulos, and C. Knauer. On some connection problems in straight-line segment arrangements. *27th EuroCG*, pages 27–30, 2011.
3. S. Bereg and D. G. Kirkpatrick. Approximating barrier resilience in wireless sensor networks. In *5th ALGOSENSORS 2009*, pages 29–40, 2009.
4. H. J. Broersma, X. Li, G. Woeginger, and S. Zhang. Paths and cycles in colored graphs. *Australasian journal of combinatorics*, 31(1):299–311, 2005.
5. D. Y. C. Chan and D. G. Kirkpatrick. Multi-path algorithms for minimum-colour path problems with applications to approximating barrier resilience. *Theor. Comput. Sci.*, 553:74–90, 2014.
6. T. M. Chan and E. Grant. Exact algorithms and apx-hardness results for geometric packing and covering problems. *Computational Geometry*, 47(2):112–124, 2014.
7. K.L. Clarkson and P.W. Shor. Application of random sampling in computational geometry, II. *Discrete & Computational Geometry*, 4:387–421, 1989.
8. I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014*, pages 624–633, 2014.
9. E. Eiben, J. Gemmell, I. Kanj, and A. Youngdahl. Improved results for minimum constraint removal. In *Proceedings of AAAI, AAAI press*, 2018.
10. E. Eiben and I. Kanj. How to navigate through obstacles? *CoRR*, abs/1712.04043, 2017. URL: `http://arxiv.org/abs/1712.04043`.
11. L. Erickson and S. LaValle. A simple, but np-hard, motion planning problem. In *Proceedings of AAAI, AAAI press*, 2013.
12. M. R. Fellows, J. Guo, and I. Kanj. The parameterized complexity of some minimum label problems. *Journal of Computer and System Sciences*, 76(8):727–740, 2010.
13. R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *J. Comb. Optim.*, 14(4):437–453, 2007.
14. K. Hauser. The minimum constraint removal problem with three robotics applications. In *Tenth Workshop on the Algorithmic Foundations of Robotics, WAFR 2012*, pages 1–17, 2012.

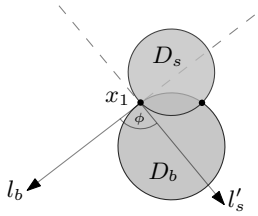

Figure 4 Angular Separation.

15 J. Hershberger, N. Kumar, and S. Suri. Shortest paths in the plane with obstacle violations. In *25th Annual European Symposium on Algorithms, (ESA 2017)*, pages 49:1–49:14, 2017.

16 K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete & Computational Geometry*, 1:59–70, 1986.

17 M. Korman, M. Löffler, R. I. Silveira, and D. Strash. On the complexity of barrier resilience for fat regions. In *9th ALGOSENSORS 2013*, pages 201–216, 2013.

18 S.Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

19 K-C. R. Tseng and D. G. Kirkpatrick. On barrier resilience of sensor networks. In *7th ALGOSENSORS 2011*, pages 130–144, 2011.

20 S. Yuan, S. Varma, and J. P. Jue. Minimum-color path problems for reliability in mesh networks. In *24th INFOCOM 2005*, volume 4, pages 2658–2669. IEEE, 2005.

A Proof of Lemma 16

To prove this lemma, we fix a small disk D_s and try to bound the number of bigger disks D_b intersected by D_s such that the tuple (x_1, D_s, D_b) was included in $\mathcal{P}$. To this end, we consider the following iterative procedure of adding tuples to $\mathcal{P}$ for a given D_s.

We process a set of *event points* $\mathcal{E}$ along the boundary of D_s in clockwise order. The event points are of the form $e_i = \langle x_1^i, D_s, D_i \rangle$ such that radius of D_i is at least the radius of D_s and D_i intersects D_s transversally at points x_1^i, x_2^i as defined before. (See also Figure 2(a).) Starting at any arbitrary point on the boundary of D_s, we sort events in $\mathcal{E}$ by its intersection point x_1^i in clockwise order. If two events $e_i, e_j \in \mathcal{E}$ have the same intersection point $x_1^i = x_1^j = x_1$, we order them by their *angular separation* from D_s at point x_1.

For any pair of intersecting disks D_s, D_b we define the angular separation at a point of intersection x_1 as follows. Consider the half line tangent l_b to disk D_b at x_1 such that l_b does not intersect D_s. Similarly, consider the half line tangent l'_s to disk D_s at x_1 but such that l'_s *intersects* D_b. The angle ϕ between l_b and l'_s is called the angular separation of D_b from D_s at the point x_1. (See also Figure 4).

Initially all disks D_i such that $e_i = \langle x_1^i, D_s, D_i \rangle$ is an event point in $\mathcal{E}$ are unmarked. Now, we simply process the events in $\mathcal{E}$ in the aforementioned order and mark the corresponding disks D_i as either *processed* or *ignored* or *discarded*. At iteration i, we consider the event point e_i and process the corresponding (unmarked) disk D_i. Now we consider the critical set $\mathcal{D}(D_s, D_i)$ as we defined before. If $|\mathcal{D}(D_s, D_i)|$ is at most k, we mark D_i as *processed* and all the unmarked disks in $\mathcal{D}(D_s, D_i)$ as *ignored*. Otherwise, we mark D_i as *discarded* and proceed to the next event point. For example, in Figure 5, we start at point x_1^1 and proceed in clockwise order. D_1 appears before D_2 in $\mathcal{E}$ because the angular separation with D_s is smaller for D_1. The disks marked as processed are shown with thick boundary and shaded gray. The disks that are marked ignored are shown in dotted and shaded orange.

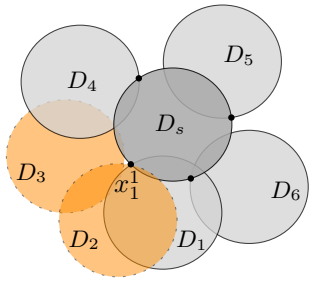

Figure 5 Processing disks D_i.

Observe that during each iteration we mark at most one disk as processed and at most k disks as ignored. Since we do not add tuples for disks that are discarded, we could have added at most $k+1$ tuples to $\mathcal{P}$ in each iteration. Next, we will show that the process terminates after a constant number of iterations. To see this, observe that the disk D_i that is marked as processed during iteration i must be exterior-disjoint w.r.t D_s from the disk D_{i-1} marked processed at the previous iteration. Thus among all these processed disks only the last disk can intersect the first one in the exterior of D_s. All other disks are exterior-disjoint of D_s from these two disks and themselves. Since all these disks are bigger than D_s, from Lemma 15 it follows that, we can have at most seven such disks around D_s. Therefore, the process must terminate after at most seven iterations and the number of tuples added to $\mathcal{P}$ corresponding to D_s is at most $7k+7$. Since there are $O(n)$ choices for D_s, we have in total $O(kn)$ candidates added to $\mathcal{P}$. ◀

B 2-Inapproximability for Rectilinear Polygons

We reduce an instance of Vertex Cover to an instance of MCR with rectilinear polygons. Recall that in the Vertex Cover problem we are given an n vertex graph $G=(V,E)$, and the goal is to find a minimum size subset $V' \subseteq V$ such that for any $(u,v) \in E$, either u or v is in V'. Let $e_1, \ldots, e_m$ be the edges of G. Next, we describe the reduction. The reduced instance of MCR contains a region called *barrier* formed by a subset of the obstacles. Each point in the barrier is contained in more than n obstacles and thus if an s-t path intersects the barrier, it intersects more than n obstacles. We would ensure that any optimal path of the instance intersects at most n obstacles and thus no such path intersects the barrier region. Intuitively, the barrier region forces any optimal path to lie in a certain region, which we refer to as *corridor*.

The construction is the following. We place an obstacle corresponding to each vertex. For each edge (u,v) there is two possible pathlets (or subpaths of an s-t path) - one that intersects the obstacle corresponding to u and the other that intersects the obstacle corresponding to v. The start (resp. end) points of the two pathlets corresponding to an edge are the same. Also one of the two pathlets corresponding to an edge lies above x-axis and the other lies below x-axis. To ensure this all the start and the end points of the pathlets are placed on the x-axis. Let s_i and t_i be the respective start and end points of the pathlets corresponding to the edge e_i. These points are placed on x-axis in the order $s_1, t_1, s_2, t_2, \ldots, s_m, t_m$. For each $1 \le i < m$, we connect the point t_i with s_{i+1} using a segment that joins the ith and $i+1^{th}$ pathlets. The point s is placed on x-axis before s_1 and t is placed on x-axis after t_m. s and s_1 are connected by a segment. Similarly, t_m and t are connected by a segment. Now to ensure that the pathlets cross the correct obstacles, they are laid out as shown in Figure 6.

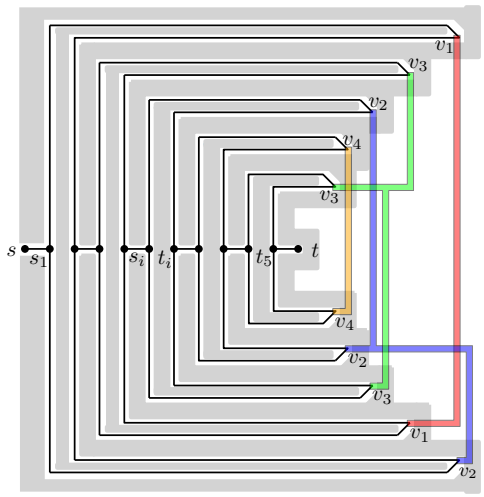

Figure 6 An example of the construction. The barrier region is shown in gray.

Each pathlet contains exactly one point (tip) having the maximum x-coordinate. Moreover, all such tips corresponding to the pathlets are in convex position. Thus one can connect the tips of any subset of pathlets using segments to form a rectilinear polygon that does not intersect any other pathlets (see Figure 6). Recall that each pathlet of an edge corresponds to a vertex. For each vertex $u \in V$, we connect the tips of the pathlets corresponding to u to form a rectilinear polygon shaped obstacle. Note that the total number of possible s-t paths we constructed is 2^m. Now to make sure that any optimal s-t path is one of these 2^m paths, we surround these paths with a barrier. Any optimal path must always stay inside the corridor, as it is expensive to cross the "wall" of the barrier. As the pathlets consist of a polynomial number of segments in total, a polynomial number of rectilinear polygons is sufficient to place avoiding the 2^m s-t paths. We make $O(n)$ copies of each such polygon to ensure the density. Lastly, each obstacle corresponding to a vertex is expanded sufficiently to ensure that it blocks the respective portion of the corridor. Note that the barrier can be placed in a way so that the corridor is arbitrarily thin, and thus this expansion can be done such that the obstacles do not cross any additional pathlets.

▶ **Lemma 22.** *There is a size k vertex cover for G iff there is an s-t path that intersects k obstacles.*

Proof. Given a cover $V' \subseteq V$ of size k for G, we simply remove the obstacles corresponding to vertices in G. Since V' covers all the edges of G, it must unblock at least one of the two pathlets corresponding to each edge (u, v) giving an obstacle-free path from s to t. Similarly given an s–t path π that intersects k obstacles, we can construct a cover for G by simply including the vertices corresponding to obstacles intersected by π. ◀

As Vertex Cover is hard to approximate within a factor of 2 assuming the Unique Games eonjecture [18], it follows that MCR with rectilinear polygons is hard to approximate within a factor of 2 (Theorem 19). It is easy to see that the same idea can easily be extended for convex polygons. Basically, one can connect the tips of any subset of pathlets using segments to form a convex polygon that does not intersect any other pathlets. This gives us the same hardness bound of 2 even for convex polygons (Corollary 20).

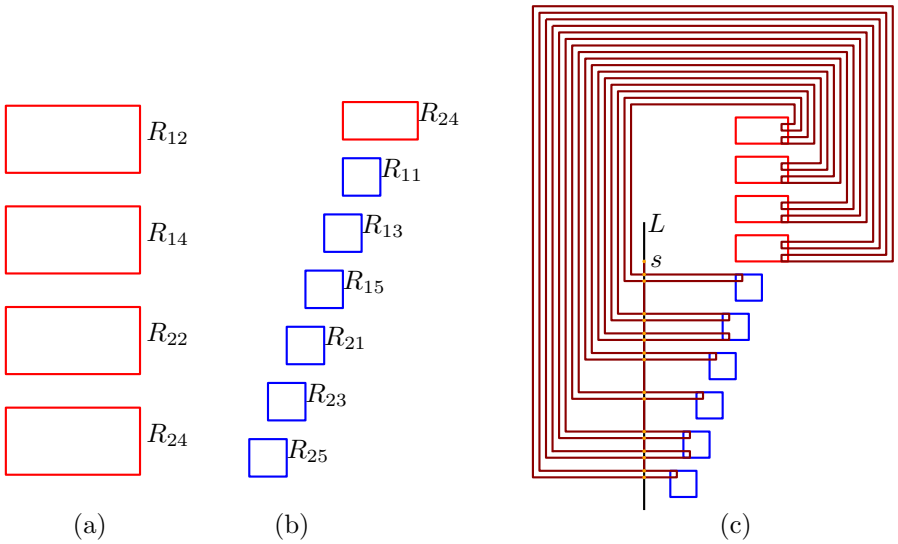

Figure 7 (a)The stack of the class 1 rectangles for $m = 2$. (b) The initial configuration of the class 2 rectangles (shown by squares) for $m = 2$. (c)Drawing of the pathlets for the class 1 edges.

C APX-hardness for Axis Parallel Rectangles

We reduce a restricted version of vertex cover to our problem which is referred to as Special-3VC. Chan et al. [6] had introduced this version for the sake of proving APX-hardness of several geometric optimization problems.

▶ **Definition 23.** In a Special-3VC instance, we are given a graph $G = (V, E)$, where V contains $5m$ vertices $\{v_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq 5\}$. E contains $4m+n$ edges - $4m$ of type 1 and n of type 2, where $2n = 3m$. Type 1 edges are of the form $\{(v_{ij}, v_{i,j+1}) \mid 1 \leq i \leq m, 1 \leq j \leq 4\}$. Type 2 edges are of the form $\{(v_{pq}, v_{xy}) \mid 1 \leq p < x \leq m,$ and q, y are odd numbers$\}$ such that any vertex v_{ij} with odd index j appears in exactly one such edge.

As each vertex v_{ij} with odd index j contributes exactly once in the type 2 edges, the number of type 2 edges is $3m/2 = n$. Chan et al. [6] proved that Special-3VC is APX-hard. Now we describe our reduction. The reduction is similar to the reduction for rectilinear polygons. We will have one obstacle corresponding to each vertex. Moreover, we construct two pathlets corresponding to each edge (u, v) such that one pathlet intersects the obstacle corresponding to u and the other intersects the obstacle corresponding to v. However, due to the simpler structure of the obstacles, here it is more complicated to show that the pathlets intersect the correct obstacles. The construction of the instance of MCR is as follows.

We denote the rectangles corresponding to v_{ij} by R_{ij}. First we place the rectangles corresponding to the vertices in $\{v_{ij} \mid 1 \leq i \leq m$ and j is even$\}$ in a way so that they form a stack like structure (see Figure 7(a)). Also the rectangles are placed from top to bottom in the lexicographic order of the indexes (i, j): R_{ab} is considered before R_{cd} if $a < c$, and R_{a2} is considered before R_{a4}. We refer to these rectangles as the class 1 rectangles. Thereafter we place the rectangles corresponding to the remaining vertices. All these rectangles are placed in the increasing order of the sum of the indexes $i + j$. The first one is placed below R_{m4} (the last rectangle of the stack) in a way so that its left side is aligned with the left side of R_{m4}. Thereafter every rectangle is placed below the already placed ones and a little aligned towards the left w.r.t. the previous one (see Figure 7(b)). We refer to these rectangles as the

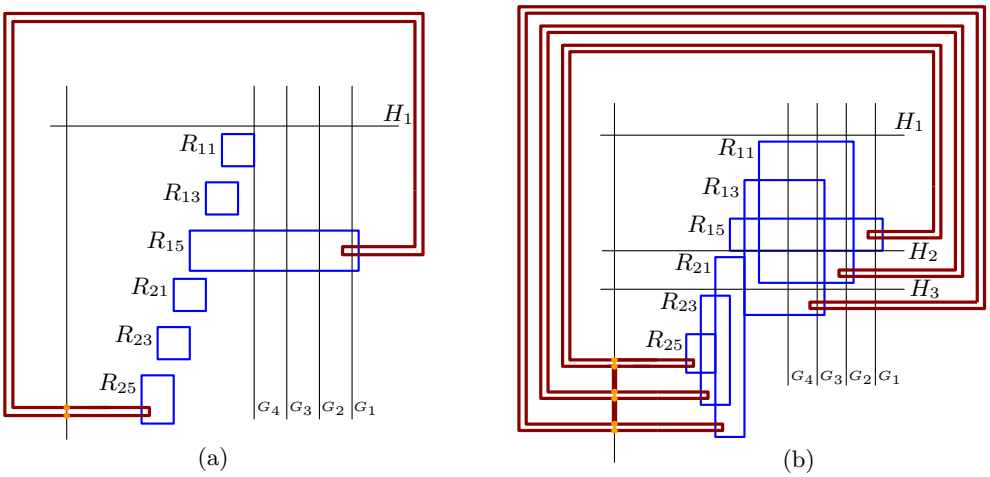

Figure 8 (a) Drawing of the pathlets for the edge (v_{pq}, v_{m5}) where $m = 2, p = 1, q = 5$. (b) Drawing of the pathlets for the type 2 edges $(v_{15}, v_{25}), (v_{11}, v_{23}), (v_{13}, v_{21})$ where $m = 2$.

class 2 rectangles. We note that initially every class 2 rectangle is a square. Later each such rectangle might be expanded suitably towards right and below to ensure the correctness of the intersections with the pathlets.

Now let L be a vertical line such that all the rectangles are placed strictly to the right of it. All the endpoints of the pathlets we draw lie on L. Each pathlet is a curve consisting of rectilinear segments. The start (resp. end) points of the two pathlets corresponding to an edge are the same. We place s right above the topmost start point of the pathlets and connect s with this point by a vertical segment. Similarly, the point t is placed below the bottommost end point and joined with it by a vertical segment. At first we draw the pathlets for type 1 edges $\{(v_{ij}, v_{i,j+1}) \mid 1 \leq i \leq m, 1 \leq j \leq 4\}$ in the dictionary order of the indexes $(i, j, i, j+1)$, i.e at first (v_{11}, v_{12}), then (v_{12}, v_{13}) and so on. The pairs of start and end points of the pathlets corresponding to these edges appear in the same order on L from top to bottom. For each type 1 edge $(v_{ij}, v_{i,j+1})$, let $s(i, j, j+1)$ and $t(i, j, j+1)$ be the respective start and end points of the pathlets. j is 3. Otherwise, it appears only once. Let P_{ij} be the horizontal projection (an interval) of R_{ij} on L. Then the start and endpoints of the pathlets of the type 1 edges with a vertex v_{ij} lie on P_{ij}. Now consider a type 1 edge $(v_{ij}, v_{i,j+1})$. Then either j or $j+1$ is odd. WLOG let j is odd. We draw the two points $s(i, j, j+1)$ and $t(i, j, j+1)$ on P_{ij} such that $s(i, j, j+1)$ lies above $t(i, j, j+1)$. One pathlet of $(v_{ij}, v_{i,j+1})$ lies on the right of L. It consists of three orthogonal segments and the only rectangle it intersects is R_{ij} (see Figure 7(c)). The other pathlet is also drawn in a way so that the only rectangle it intersects is $R_{i,j+1}$ (see Figure 7(c)). We repeat the process for all type 1 edges and each consecutive pairs of end and start points are joined with a vertical segment.

Now we draw the pathlets corresponding to the type 2 edges $\{(v_{pq}, v_{xy}) \mid 1 \leq p < x \leq m,$ and q, y are odd numbers$\}$. Note that, there are n such edges in G. We process all these edges in the reverse lexicographic order of the indexes (x, y) of the vertices v_{xy}. Thus at first we consider the edge that contains v_{m5}, then the edge that contains v_{m3} (if not considered already), then the edge that contains $v_{m-1,5}$ (if not considered already), and so on. We take $n+1$ vertical lines $G_1, \ldots, G_{n+1}$ such that G_{n+1} intersects the right vertical side of R_{11}, G_n is on the right of G_{n+1}, G_{n-1} is on the right of G_n, and in general G_i is on the right of G_{i+1}. Also let G_i and G_{i+1} are unit distance apart for $1 \leq i \leq n$. In every iteration $1 \leq i \leq n$, we define a horizontal line H_i. Denote the region by Q_i, that lie below H_i and inside the

strip defined by G_i and G_{i+1}. The drawing procedure is the following. Consider the first edge (v_{pq}, v_{m5}) corresponding to the vertex v_{m5}. Let H_1 be a horizontal line such that all the class 1 rectangles lie above it and all the class 2 rectangles lie below it. At first we expand R_{m5} sufficiently towards below such that one can place a pathlet with the following properties - the only rectangle it intersects is R_{m5}, it consists of two horizontal segments and one vertical segment, and its start and end points lie on L. Note that the expansion of R_{m5} do not create any new intersections with the existing pathlets. Thereafter R_{pq} is expanded sufficiently towards below and right to ensure that it has non-empty intersection with Q_1. Then the other pathlet can be drawn in a way so that it intersects the portion of R_{pq} that is in Q_1, and as Q_1 is empty the pathlet does not intersect any other rectangle (see Figure 8(a)). Now consider the ith type 2 edge (v_{pq}, v_{xy}) in this order. Let all the edges before it in the ordering are already taken care of. It is easy to see that one can expand R_{xy} towards below for drawing a pathlet with the desired properties. Now to make sure that the other pathlet intersects only R_{pq}, set H_i to be a horizontal line such that the region Q_i, as defined above, is empty of previously drawn pathlets and expanded rectangles. Then we can expand R_{pq} towards below and right so that it has non-empty intersection with Q_i. As Q_i is empty one can draw the other pathlet as well with the desired properties (see Figure 8(b)).

Finally, we place the barrier region around the paths. As the pathlets are orthogonal and consisting of a polynomial number of segments in total, the barrier region can be simulated using a polynomial number of rectangles and thus the construction can be realized in polynomial time.

From the construction, it is straightforward to see the following lemma.

▶ **Lemma 24.** *There is a size k vertex cover for G iff there is an s-t path that intersects k rectangles.*

As Special-3VC is APX-hard, it follows that MCR with axis-aligned rectangles is APX-hard (Theorem 21).