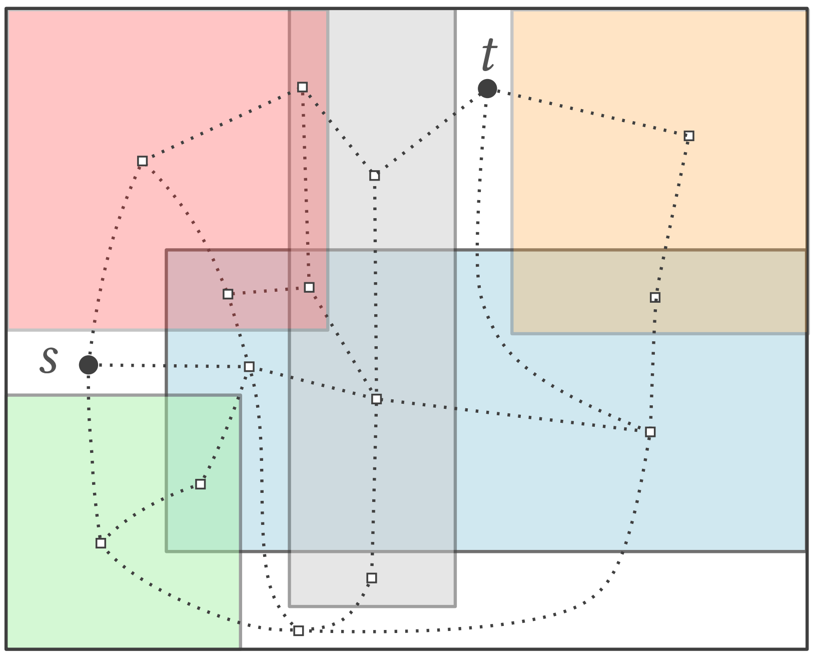
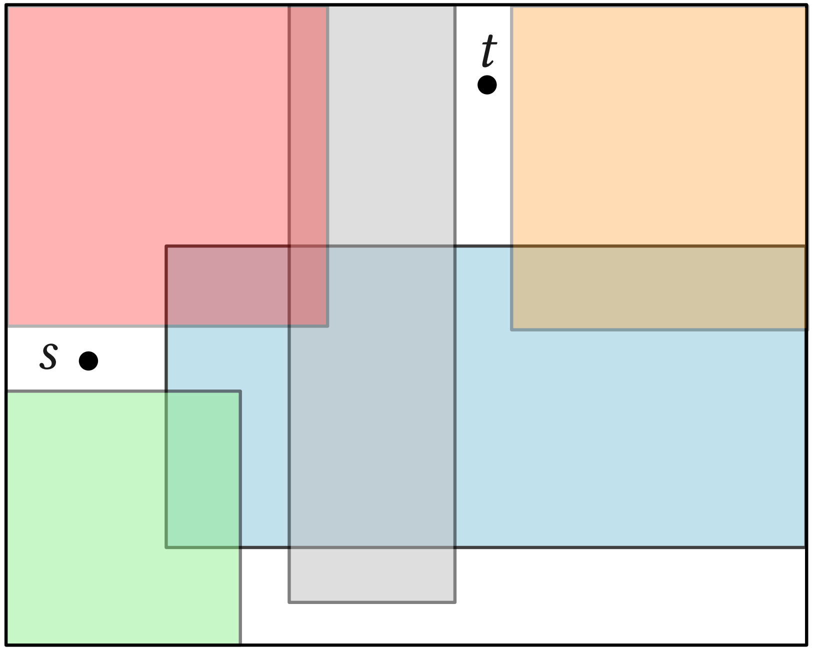


Improved Approximation Bounds for the Minimum Constraint Removal Problem

Sayan Bandyapadhyay, Neeraj Kumar, Subhash Suri and Kasturi Varadrajan

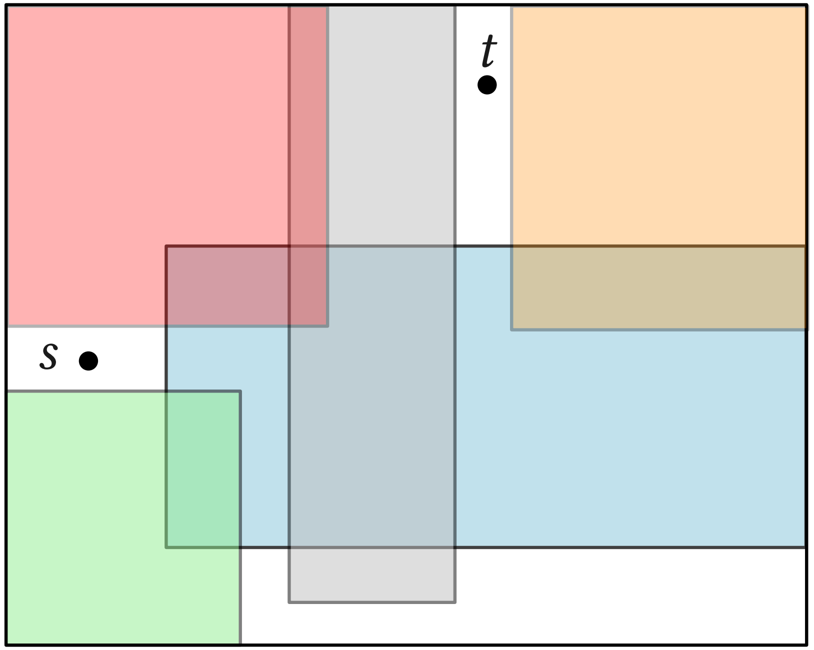
University of Iowa and UC Santa Barbara

Problem Description



Input : An arrangement of obstacles in plane, source s, target t

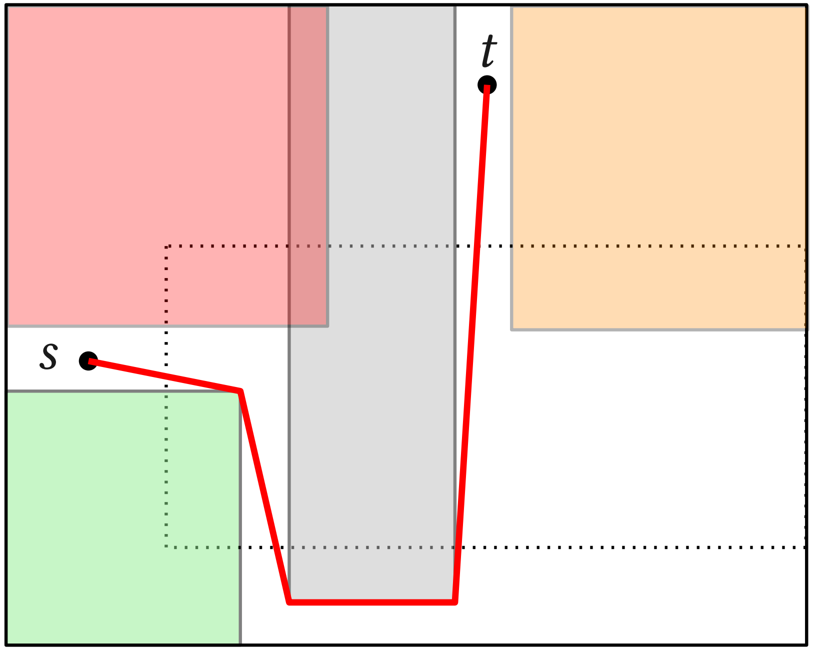
Problem Description



Input : An arrangement of obstacles in plane, source s, target t

Find the smallest set of obstacles whose removal gives an obstacle free s–t path

Problem Description



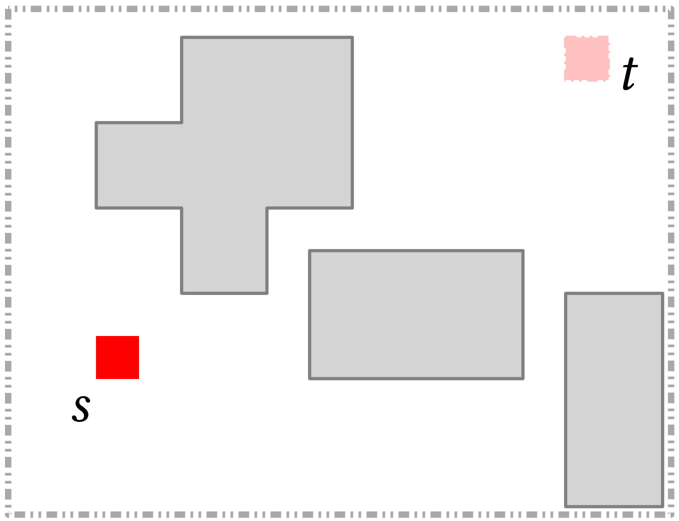
Input : An arrangement of obstacles in plane, source s, target t

Find the smallest set of obstacles whose removal gives an obstacle free s–t path

Motivation

- **Robot Motion Planning**

  Configuration space approach:

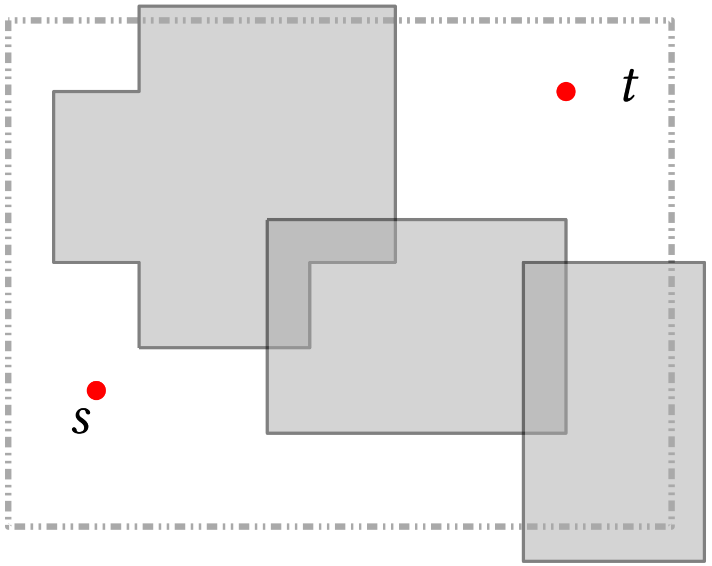


Motivation

- **Robot Motion Planning**

  Configuration space approach:

  – Shrinks robot to a point, expands obstacles

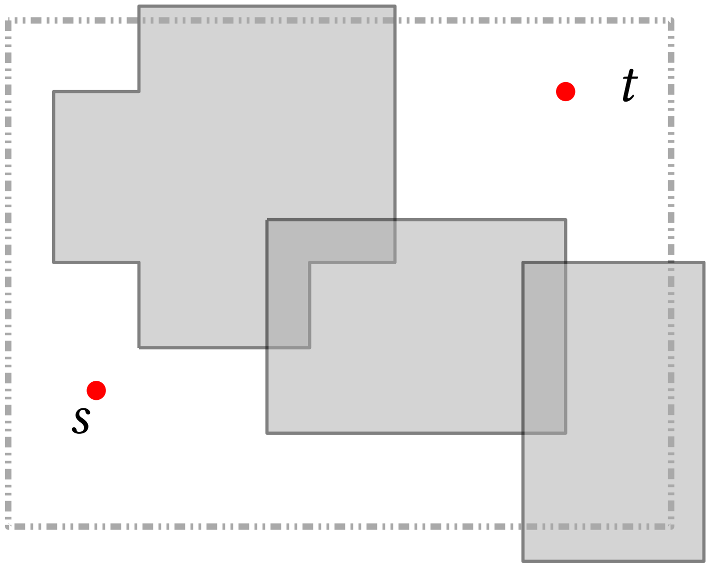


Motivation

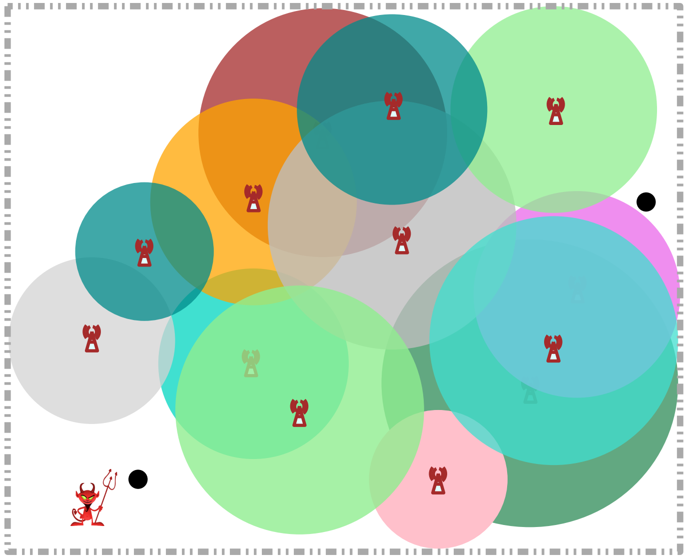
- **Robot Motion Planning**

  Configuration space approach:

  – Shrinks robot to a point, expands obstacles



- **Sensor Network Coverage**
  (commonly known as *Barrier Resillience*)

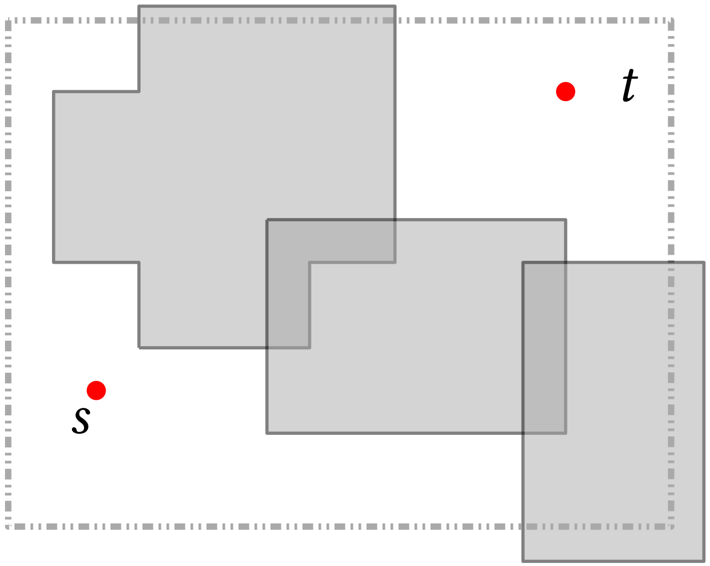


Motivation

- **Robot Motion Planning**

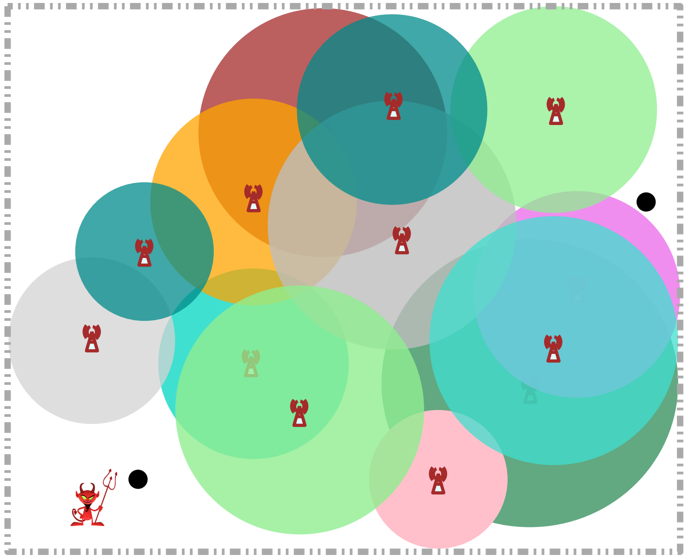
  Configuration space approach:

  – Shrinks robot to a point, expands obstacles



- **Sensor Network Coverage**
  (commonly known as *Barrier Resillience*)

  Which sensors to disable?

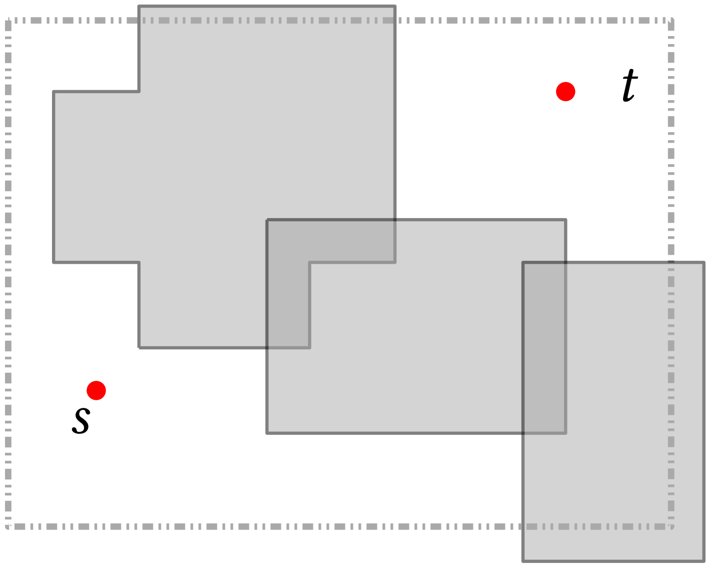


Motivation

- **Robot Motion Planning**

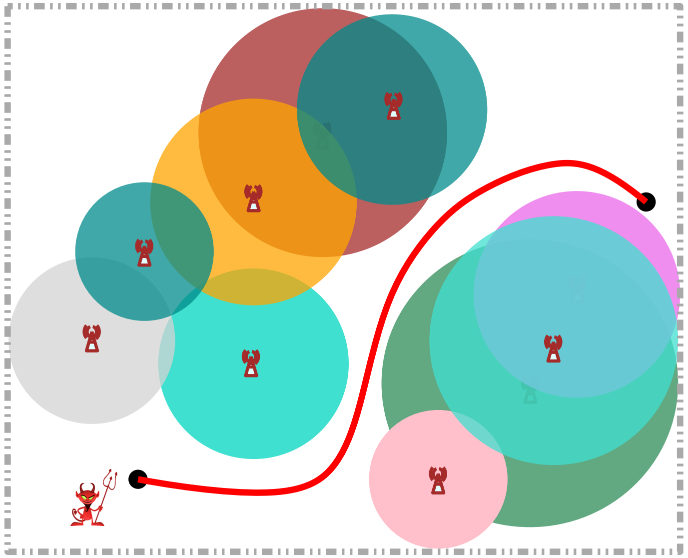
  Configuration space approach:

  – Shrinks robot to a point, expands obstacles



- **Sensor Network Coverage**
  (commonly known as *Barrier Resillience*)

  Which sensors to disable?



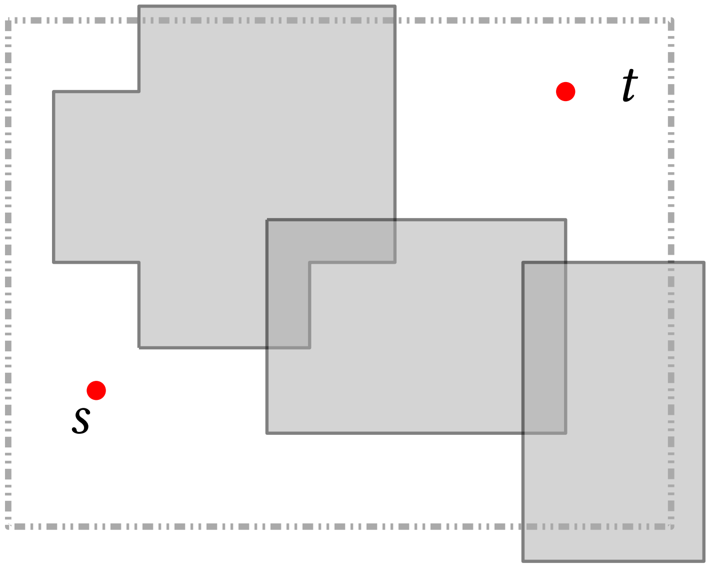
Motivation

- **Robot Motion Planning**

  Configuration space approach:

  – Shrinks robot to a point, expands obstacles

  [Polygonal Obstacles]

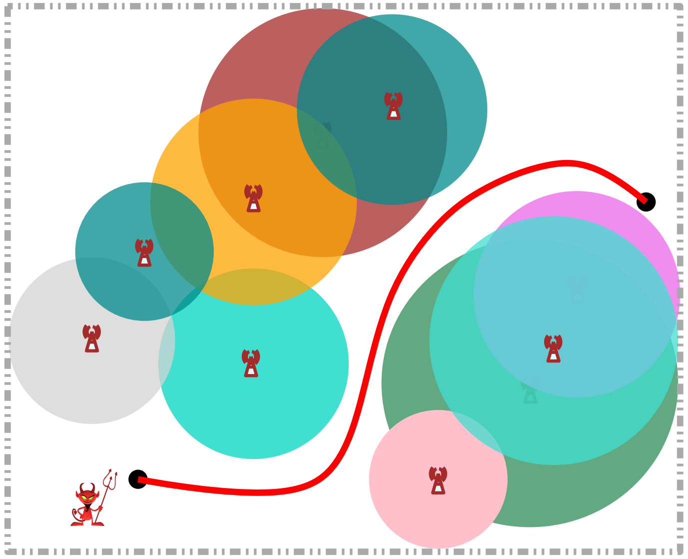


- **Sensor Network Coverage**
  (commonly known as *Barrier Resillience*)

  Which sensors to disable?

  Barrier Resillience ≡ Minimum number of sensors to disable

  [Disk Obstacles]



Previous Work

- Known to be intractable but no known approximations

  [Erickson and LaValle, 2013, Eiben et al. 2018]

Previous Work

- Known to be intractable but no known approximations

  [Erickson and LaValle, 2013, Eiben et al. 2018]

- Barrier resillience extensively studied in sensor network community
  - Constant approximation for unit disks and few other restricted cases

    [Bereg and Kirkpatrick, 2009, Chan and Kirkpatrick, 2014]

Previous Work

- Known to be intractable but no known approximations

  [Erickson and LaValle, 2013, Eiben et al. 2018]

- Barrier resillience extensively studied in sensor network community
  - Constant approximation for unit disks and few other restricted cases

  [Bereg and Kirkpatrick, 2009, Chan and Kirkpatrick, 2014]

Key Idea: Optimal path does not cross an obstacle a lot of times

Previous Work

- Known to be intractable but no known approximations

  [Erickson and LaValle, 2013, Eiben et al. 2018]

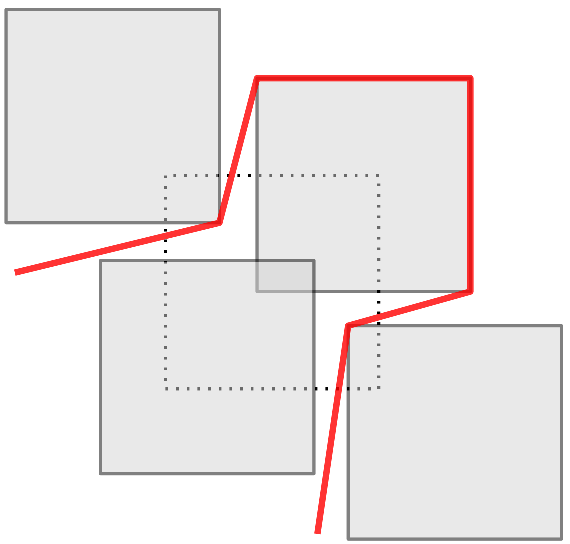
- Barrier resillience extensively studied in sensor network community
  – Constant approximation for unit disks and few other restricted cases

  [Bereg and Kirkpatrick, 2009, Chan and Kirkpatrick, 2014]

Key Idea: Optimal path does not cross an obstacle a lot of times

For unit square obstacles, this number is 2

– 2-approximation readily follows



Our Results

- An $O(\sqrt{n})$-approximation if input obstacles are rectilinear polygons
- An $O(\sqrt{n})$-approximation for arbitrary disk obstacles
- An $O(\sqrt{n}\alpha(n))$-approximation for arbitrary polygons

  Inverse Ackerman function

Our Results

- An $O(\sqrt{n})$-approximation if input obstacles are rectilinear polygons
- An $O(\sqrt{n})$-approximation for arbitrary disk obstacles
- An $O(\sqrt{n}\alpha(n))$-approximation for arbitrary polygons

  Inverse Ackerman function
- **Hardness of Approximation**
  - For rectilinear obstacles, hard to approximate within a factor of 2
  - For rectangles, the problem is APX-hard

Our Results

- An $O(\sqrt{n})$-approximation if input obstacles are rectilinear polygons
- An $O(\sqrt{n})$-approximation for arbitrary disk obstacles
- An $O(\sqrt{n}\alpha(n))$-approximation for arbitrary polygons

  Inverse Ackerman function

- **Hardness of Approximation**
  - For rectilinear obstacles, hard to approximate within a factor of 2
  - For rectangles, the problem is APX-hard

→ Based on an **algorithmic framework** that also gives

- An $O(\sqrt{n})$-approximation for *min-color path* problem in graphs

Our Results

- An $O(\sqrt{n})$-approximation if input obstacles are rectilinear polygons
- An $O(\sqrt{n})$-approximation for arbitrary disk obstacles
- An $O(\sqrt{n}\alpha(n))$-approximation for arbitrary polygons

  Inverse Ackerman function

- **Hardness of Approximation**
  - For rectilinear obstacles, hard to approximate within a factor of 2
  - For rectangles, the problem is APX-hard

Based on an **algorithmic framework** that also gives

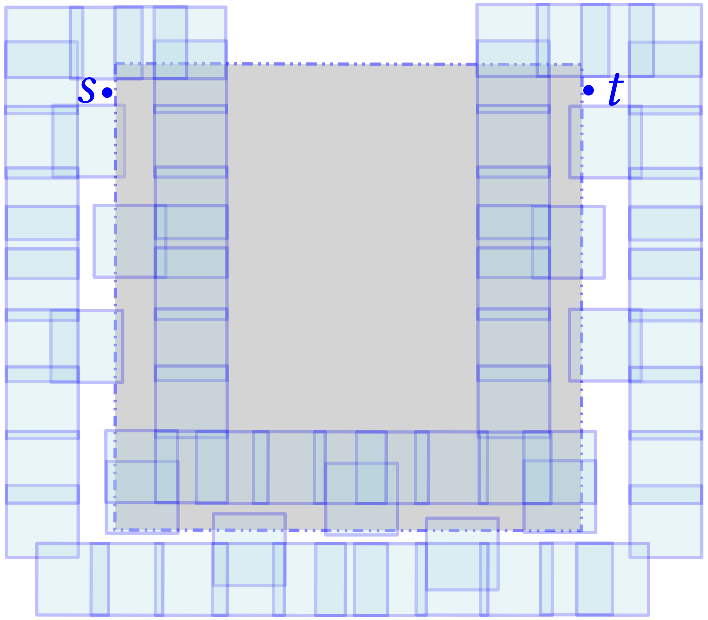
- An $O(\sqrt{n})$-approximation for *min-color path* problem in graphs

Rest of this talk

Main Idea

Previous work: Optimal path does not cross an obstacle a lot of times

For more general obstacles:



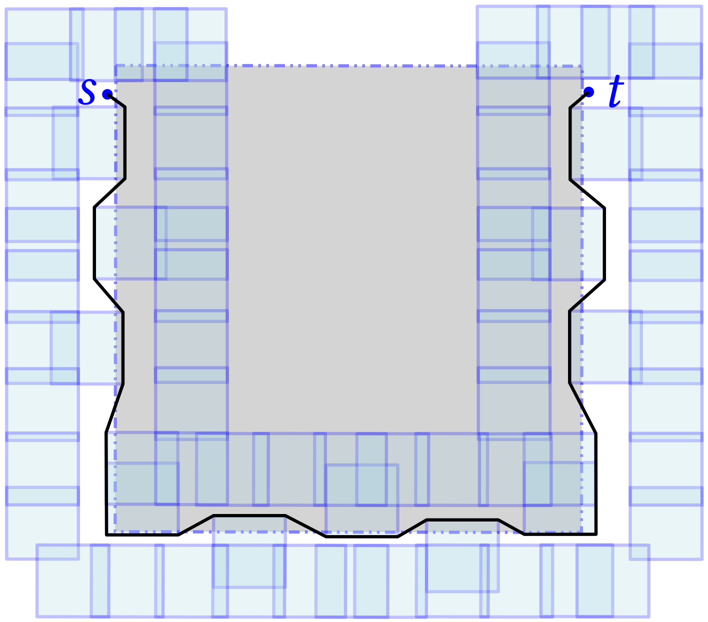
Even for axis-aligned squares

Main Idea

Previous work: Optimal path does not cross an obstacle a lot of times

For more general obstacles:

– The optimal path can cross the same obstacle up to $\Theta(n)$ times



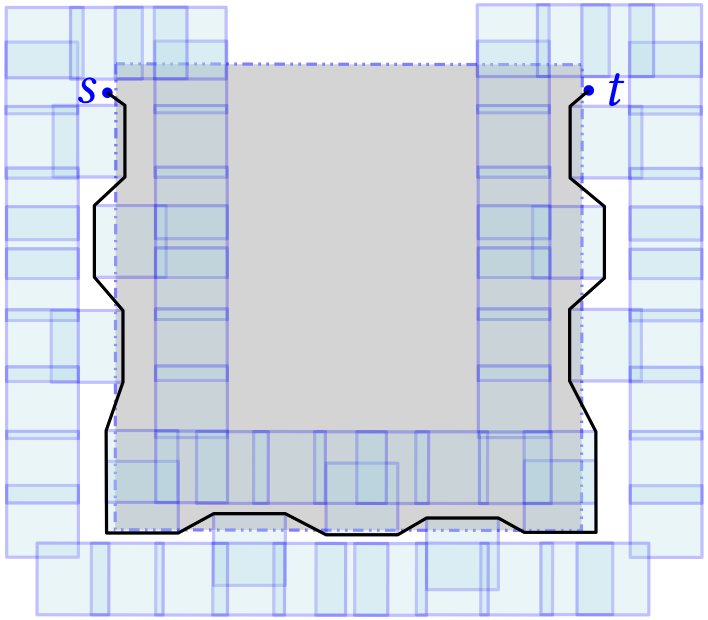
Even for axis-aligned squares

Main Idea

Previous work: Optimal path does not cross an obstacle a lot of times

For more general obstacles:

– The optimal path can cross the same obstacle up to $\Theta(n)$ times



Even for axis-aligned squares

To alleviate this problem:

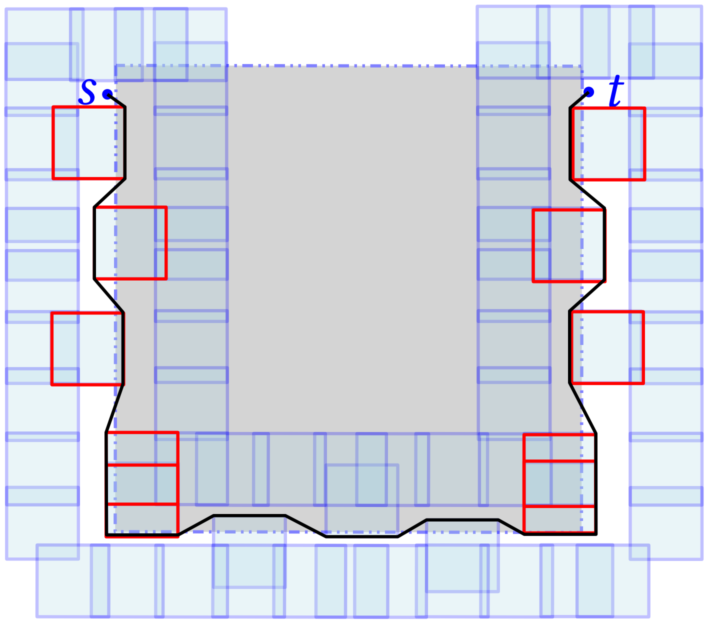
– we introduce a **filtering step**

Main Idea

Previous work: Optimal path does not cross an obstacle a lot of times

For more general obstacles:

– The optimal path can cross the same obstacle up to $\Theta(n)$ times



Even for axis-aligned squares

To alleviate this problem:

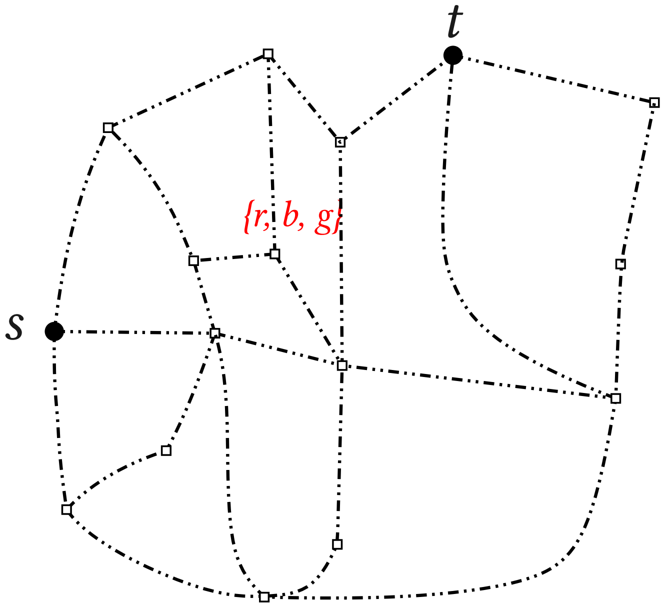
– we introduce a **filtering step**

Identify a “small set” of potentially dangerous obstacles

A Related Graph Problem

Input: Graph $G = (V, E)$, set of colors $\mathcal{C}$

Every vertex is assigned a subset $\mathcal{X}(v) \subseteq \mathcal{C}$

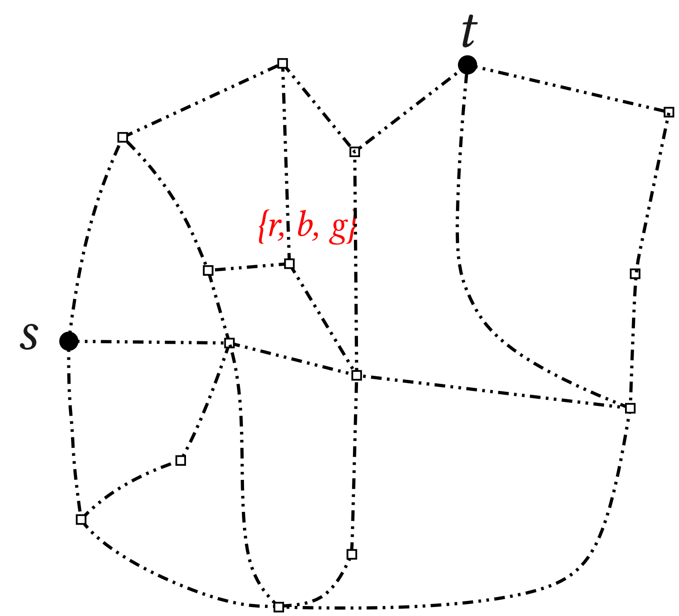


A Related Graph Problem

Input: Graph $G = (V, E)$, set of colors $\mathcal{C}$

Every vertex is assigned a subset $\mathcal{X}(v) \subseteq \mathcal{C}$

Find the minimum-color path from s to t



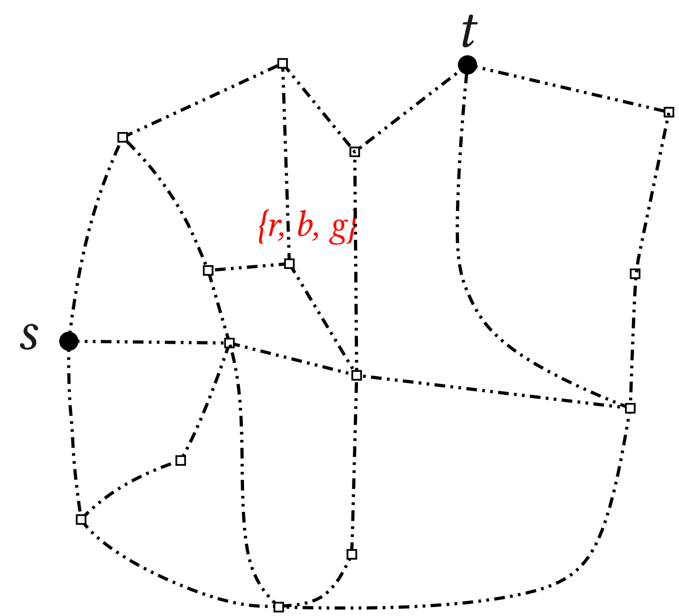
A Related Graph Problem

Input: Graph $G = (V, E)$, set of colors $\mathcal{C}$

Every vertex is assigned a subset $\mathcal{X}(v) \subseteq \mathcal{C}$

Find the minimum-color path from s to t

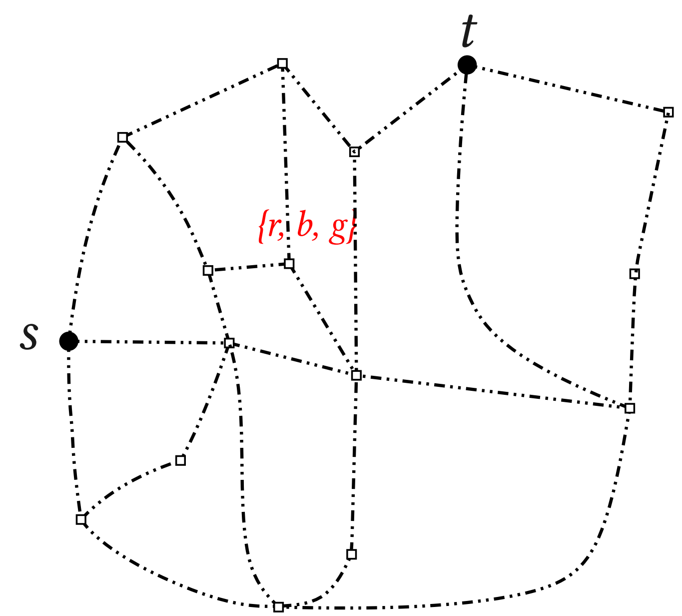
that is, $|\bigcup_{v \in path} \mathcal{X}(v)|$ is minimum



A Related Graph Problem

Input: Graph $G = (V, E)$, set of colors $\mathcal{C}$

Every vertex is assigned a subset $\mathcal{X}(v) \subseteq \mathcal{C}$



Find the minimum-color path from s to t

that is, $|\bigcup_{v \in path} \mathcal{X}(v)|$ is minimum

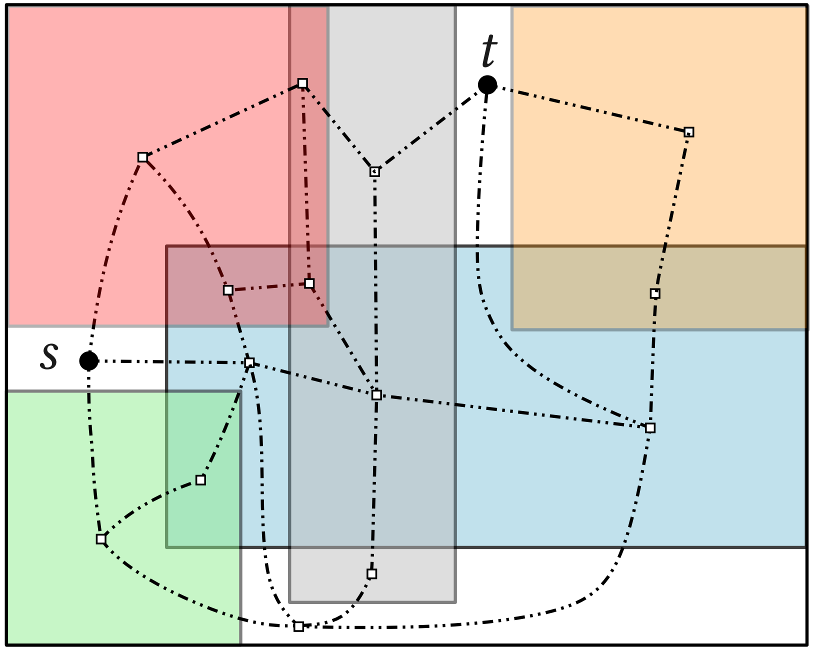
Hard to approximate within a logarithmic factor

(simple reduction from Set-Cover)

A Related Graph Problem

Input: Graph $G = (V, E)$, set of colors $\mathcal{C}$

Every vertex is assigned a subset $\mathcal{X}(v) \subseteq \mathcal{C}$



Find the minimum-color path from s to t

that is, $|\bigcup_{v \in path} \mathcal{X}(v)|$ is minimum

Hard to approximate within a logarithmic factor

(simple reduction from Set-Cover)

Minimum Constraint removal can be cast as an instance of min-color path

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \rightarrow 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$

can be any arbitrary set

that satisfies:

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \rightarrow 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$ **that satisfies:**

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

In other words, C appears on path "because of" an element in $\mathcal{N}(C)$

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \rightarrow 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$

that satisfies:

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

Algorithm Outline

– Construct neigborhood $\mathcal{N}$, discard colors C that have $|\mathcal{N}(C)| > \sqrt{n}$

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \to 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$ **that satisfies:**

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

Algorithm Outline

– Construct neigborhood $\mathcal{N}$, discard colors C that have $|\mathcal{N}(C)| > \sqrt{n}$

– Assign cardinalities as weight and return the shortest s-t path

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \to 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$ **that satisfies:**

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

$O(k\sqrt{n})$ discards

Algorithm Outline

– Construct neigborhood $\mathcal{N}$, discard colors C that have $|\mathcal{N}(C)| > \sqrt{n}$

– Assign cardinalities as weight and return the shortest s-t path

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \to 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$ **that satisfies:**

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

$O(k\sqrt{n})$ discards

Algorithm Outline

– Construct neigborhood $\mathcal{N}$, discard colors C that have $|\mathcal{N}(C)| > \sqrt{n}$

– Assign cardinalities as weight and return the shortest s-t path

uses $O(k\sqrt{n})$ colors

An Approximation Framework

Input: A colored graph $G = (V, E, \mathcal{C})$, and an integer k find a k-color path

Key Idea: Define a notion of neighborhood $\mathcal{N}$ for colors in $\mathcal{C}$

- Discard colors with 'dense' neighborhoods

Neighborhood $\mathcal{N} : C \to 2^{\mathcal{P}}$ is a mapping from colors to a subset of objects $\mathcal{P}$ **that satisfies:**

- *Bounded-size* : Total neighborhood size is $O(kn)$
- *Bounded-occurrence:* A color C appears on path at most $O(|\mathcal{N}(C)|)$ times

$O(k\sqrt{n})$ discards

Algorithm Outline

– Construct neigborhood $\mathcal{N}$, discard colors C that have $|\mathcal{N}(C)| > \sqrt{n}$

– Assign cardinalities as weight and return the shortest s-t path

uses $O(k\sqrt{n})$ colors

$O(\sqrt{n})$-**approximation!**

$O(\sqrt{|V|})$-approximation for Min-Color Path

Suffices to just find the neighborhood $\mathcal{N}$

$O(\sqrt{|V|})$-approximation for Min-Color Path

Suffices to just find the neighborhood $\mathcal{N}$

$\mathcal{P} = V$, the vertices of colored-graph $G = (V, E, \mathcal{C})$

$\mathcal{N}(C)$ = vertices that contain C and have at most k colors

$O(\sqrt{|V|})$-approximation for Min-Color Path

Suffices to just find the neighborhood $\mathcal{N}$

$\mathcal{P} = V$, the vertices of colored-graph $G = (V, E, \mathcal{C})$

$\mathcal{N}(C)$ = vertices that contain C and have at most k colors

Can easily verify bounded size and occurrence properties

$\sum_{C \in \mathcal{C}} \mathcal{N}(C) = O(k|V|)$

$O(\sqrt{n})$-approximation

$O(\sqrt{|V|})$-approximation for Min-Color Path

Suffices to just find the neighborhood $\mathcal{N}$

$\mathcal{P} = V$, the vertices of colored-graph $G = (V, E, \mathcal{C})$

$\mathcal{N}(C)$ = vertices that contain C and have at most k colors

Can easily verify bounded size and occurrence properties

$\sum_{C \in \mathcal{C}} \mathcal{N}(C) = O(k|V|)$

$O(\sqrt{n})$**-approximation**

Does not really help with minimum constraint removal as $|V| = \Omega(n^2)$

With more effort, can still find a sparse neighborhood $\mathcal{N}$

Application to Geometric Objects

Step 1. Represent input as a colored graph $G = (V, E, \mathcal{C})$ such that:

- k-color path in G corresponds to path removing $\leq k$ obstacles
- path removing k obstacles corresponds to $\leq k$-color path

Application to Geometric Objects

Step 1. Represent input as a colored graph $G = (V, E, \mathcal{C})$ such that:

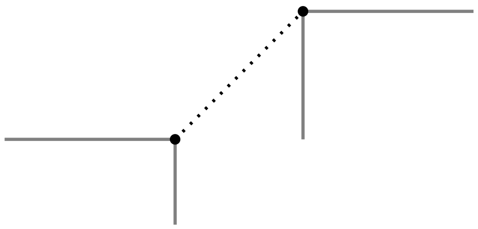
- k-color path in G corresponds to path removing $\leq k$ obstacles
- path removing k obstacles corresponds to $\leq k$-color path

Step 2. Construct the neighborhood $\mathcal{N}$

Rectilinear Polygons

Graph Construction

$G = (V, E)$: complete graph over all n polygon vertices

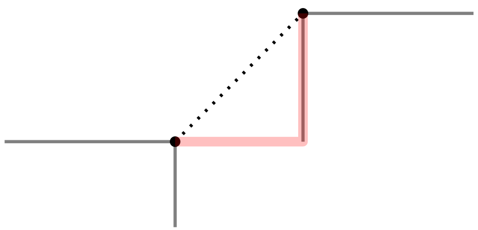


To ensure k-color path in G corresponds to path removing $\leq k$ obstacles

Rectilinear Polygons

Graph Construction

$G = (V, E)$: complete graph over all n polygon vertices



L-shaped embeddings of edges

To ensure k-color path in G corresponds to path removing $\leq k$ obstacles

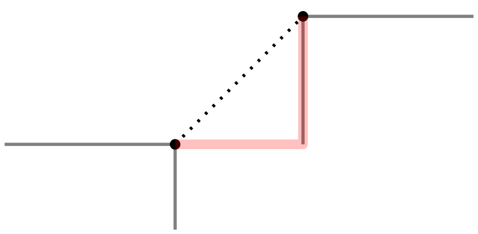
– Fix embedding e_{uv} of an edge (u, v)

– Assign colors corresponding to obstacles crossed by e_{uv}

Rectilinear Polygons

Graph Construction

$G = (V, E)$: complete graph over all n polygon vertices



L-shaped embeddings of edges

To ensure k-color path in G corresponds to path removing $\leq k$ obstacles

– Fix embedding e_{uv} of an edge (u, v)

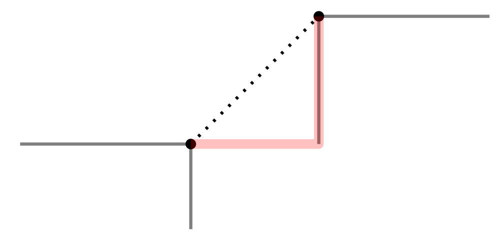
– Assign colors corresponding to obstacles crossed by e_{uv}

To ensure path removing k obstacles corresponds to $\leq k$-color path in G

Can show: path crossing k-obstacles can be made rectilinear
without crossing extra obstacles

Rectilinear Polygons

Graph Construction



L-shaped embeddings of edges

$G = (V, E)$: complete graph over all n polygon vertices

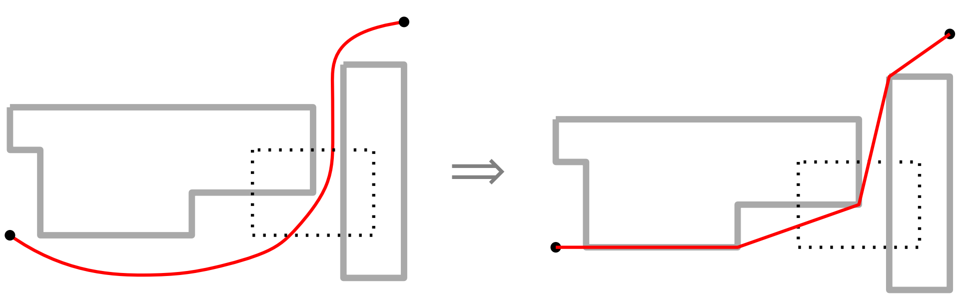
To ensure k-color path in G corresponds to path removing $\leq k$ obstacles

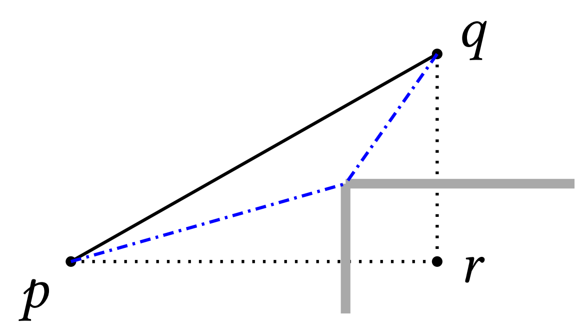
– Fix embedding e_{uv} of an edge (u, v)

– Assign colors corresponding to obstacles crossed by e_{uv}

To ensure path removing k obstacles corresponds to $\leq k$-color path in G

Can show: path crossing k-obstacles can be made rectilinear without crossing extra obstacles



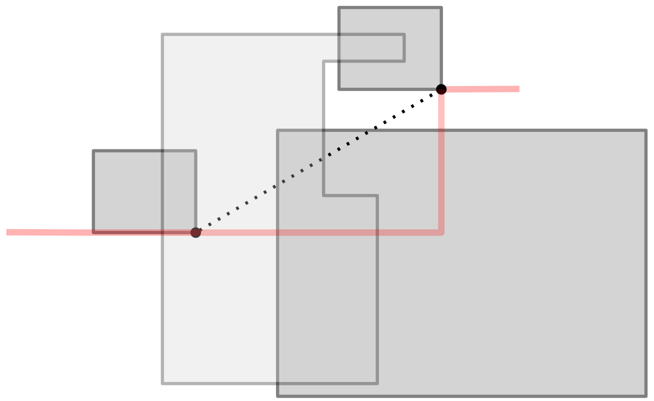


Rectilinear Polygons

Neighborhood Construction

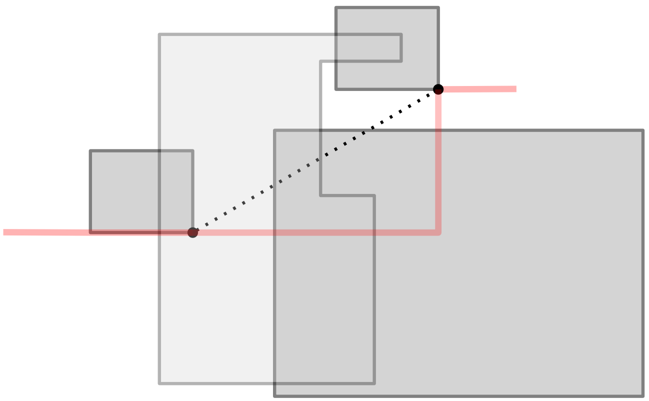
What causes the path to cross a given obstacle?

–An obstacle corner



Rectilinear Polygons

Neighborhood Construction



What causes the path to cross a given obstacle?

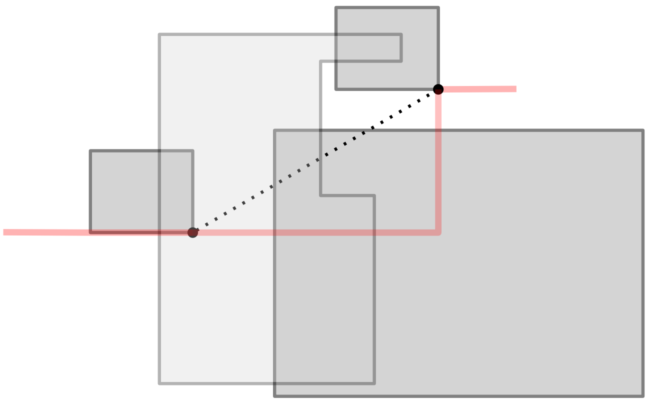
–An obstacle corner

Obvious candidate for the set $\mathcal{P}$: set of all corners

How to add corners to $\mathcal{N}(C)$ of obstacle C ensuring small total size?

Rectilinear Polygons

Neighborhood Construction

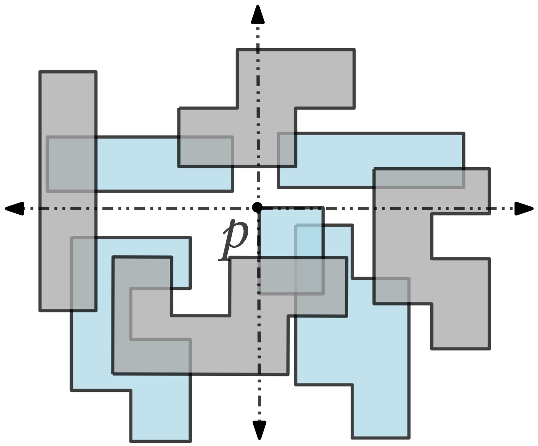


What causes the path to cross a given obstacle?

–An obstacle corner

Obvious candidate for the set $\mathcal{P}$: set of all corners

How to add corners to $\mathcal{N}(C)$ of obstacle C ensuring small total size?



– Shoot a ray at a corners p

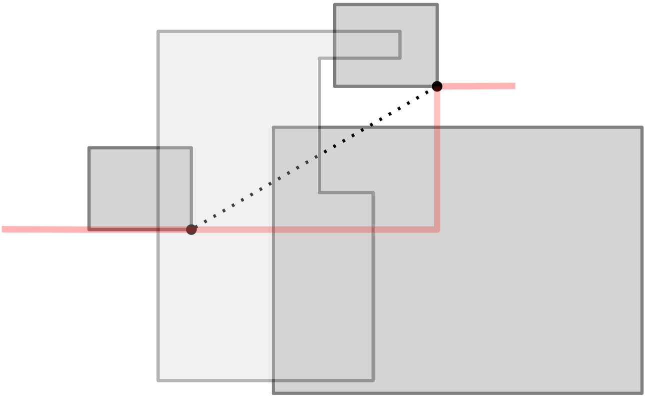
– Include p to neighborhood of first k obstacles

Rectilinear Polygons

Neighborhood Construction

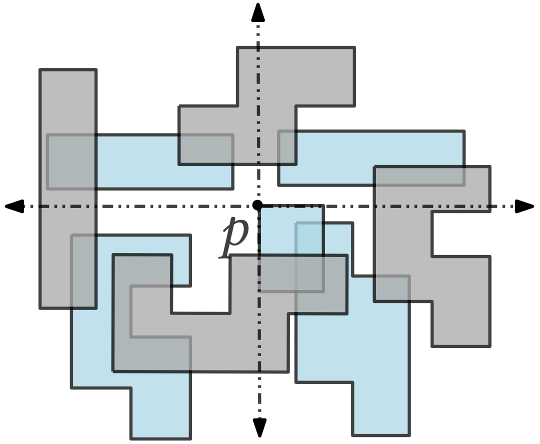
What causes the path to cross a given obstacle?

–An obstacle corner



Obvious candidate for the set $\mathcal{P}$: set of all corners

How to add corners to $\mathcal{N}(C)$ of obstacle C ensuring small total size?



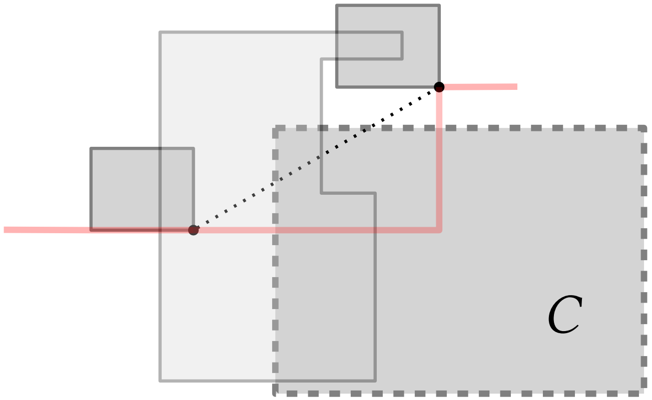
– Shoot a ray at a corners p

– Include p to neighborhood of first k obstacles

Total neighborhood size = $4k \cdot n$

Rectilinear Polygons

Each crossing of C by a valid k-color path is charged to a *neighbor* corner

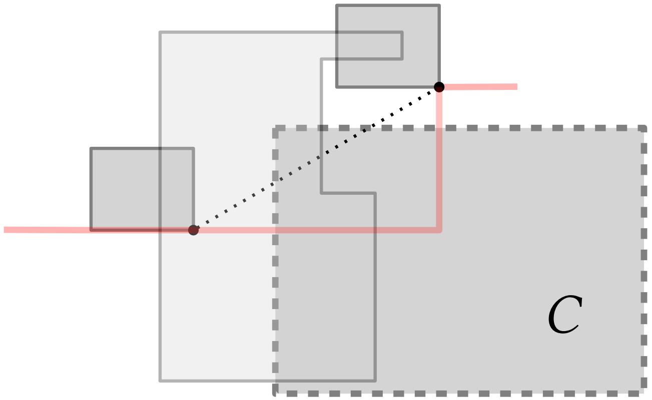


Rectilinear Polygons

Each crossing of C by a valid k-color path is charged to a *neighbor* corner

Each corner is charged for C at most twice

C is crossed $O(|\mathcal{N}(C)|)$ times

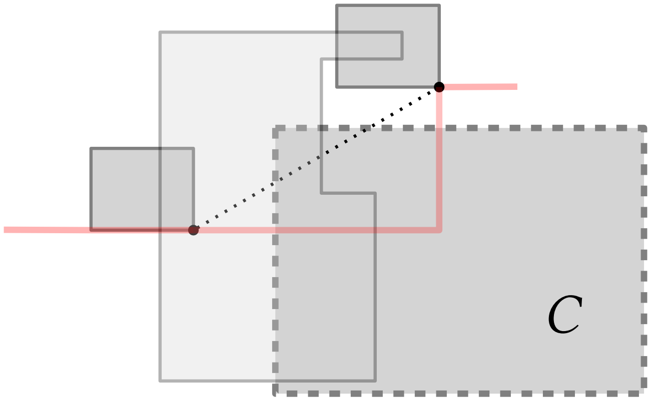


Rectilinear Polygons

Each crossing of C by a valid k-color path is charged to a *neighbor* corner

Each corner is charged for C at most twice

C is crossed $O(|\mathcal{N}(C)|)$ times



Both bounded size and bounded occurrence propertiey are satisfied

$O(\sqrt{n})$**-approximation!**

Concluding Remarks

- First sublinear approximation for minimum constraint removal problem
  - Almost $O(\sqrt{n})$ for all geometric objects

Concluding Remarks

- First sublinear approximation for minimum constraint removal problem
  - Almost $O(\sqrt{n})$ for all geometric objects
- Improved hardness of approximation results
  - Factor of 2 for rectilinear/convex polygons, APX-hardness with rectangles

Concluding Remarks

- First sublinear approximation for minimum constraint removal problem
  - Almost $O(\sqrt{n})$ for all geometric objects
- Improved hardness of approximation results
  - Factor of 2 for rectilinear/convex polygons, APX-hardness with rectangles
- First sublinear approximation for min-color path

  Problem also recently studied under FPT lenses for some special graph classes [Eiben and Kanj, ICALP'18]

Concluding Remarks

- First sublinear approximation for minimum constraint removal problem
  - Almost $O(\sqrt{n})$ for all geometric objects
- Improved hardness of approximation results
  - Factor of 2 for rectilinear/convex polygons, APX-hardness with rectangles
- First sublinear approximation for min-color path

Problem also recently studied under FPT lenses for some special graph classes [Eiben and Kanj, ICALP'18]

Natural open problem : Can we improve upon these bounds?

Likely to need new techniques ...

Concluding Remarks

- First sublinear approximation for minimum constraint removal problem
  - Almost $O(\sqrt{n})$ for all geometric objects
- Improved hardness of approximation results
  - Factor of 2 for rectilinear/convex polygons, APX-hardness with rectangles
- First sublinear approximation for min-color path

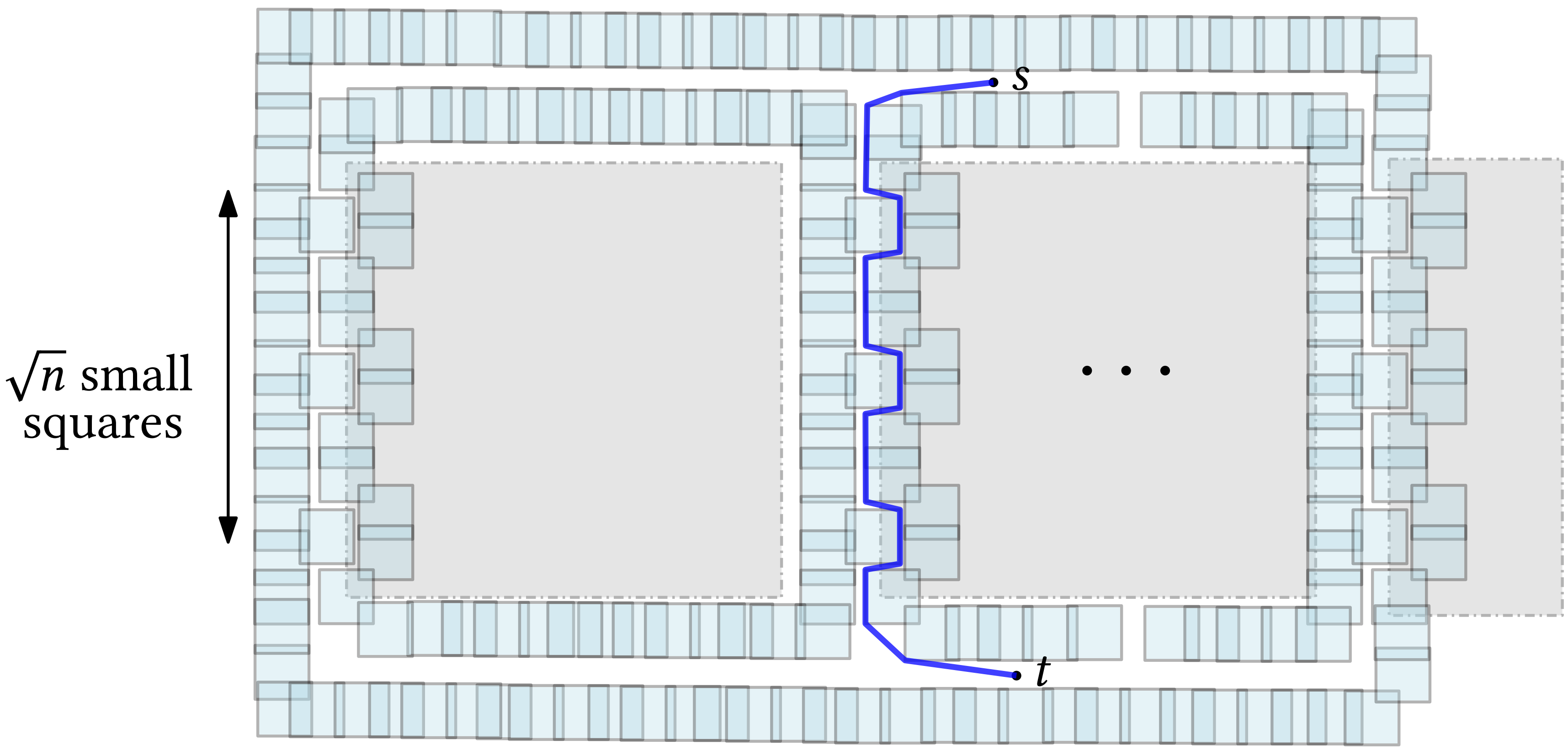
  Problem also recently studied under FPT lenses for some special graph classes [Eiben and Kanj, ICALP'18]

Natural open problem : Can we improve upon these bounds?

Likely to need new techniques ...

Thanks!

Backup : Tight Example



Create $\Theta(n)$ such channels