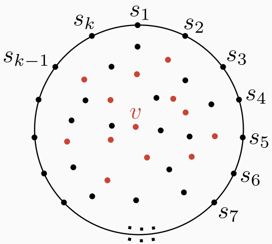
Improved Compression of the Okamura-Seymour Metric

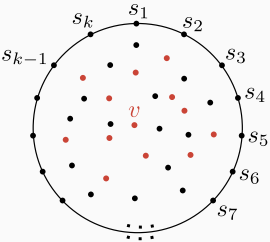
Shay Mozes, Nathan Wallheimer, Oren Weimann

The Okamura-Seymour Metric Compression Problem



- An undirected, unweighted planar graph $G = (V, E)$.
- A set $S = \{s_1, s_2, \ldots, s_k\}$ of k consecutive vertices on a face f_∞.
- A set $T \subseteq V$ of terminal vertices lying anywhere in the graph.

The Okamura-Seymour Metric Compression Problem

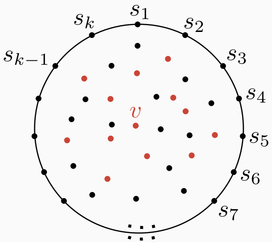


- An undirected, unweighted planar graph $G = (V, E)$.
- A set $S = \{s_1, s_2, \ldots, s_k\}$ of k consecutive vertices on a face f_∞.
- A set $T \subseteq V$ of terminal vertices lying anywhere in the graph.

The Okamura-Seymour Metric Compression Problem

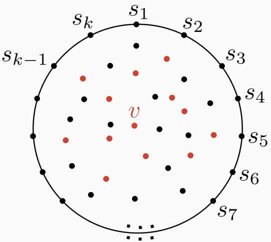
Succinctly encode the $T \times S$ distances to answer $d(v, s_i)$ queries.

Results



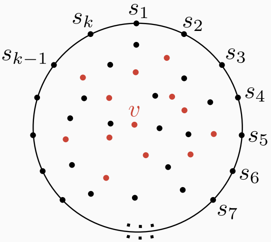
	Solution	Complexity
$T = S$	Unit-Monge [AGMW'18]	$\tilde{O}(k)$ space, $\tilde{O}(1)$ query
$T = V$	MSSP [Klein'05]	$O(n)$ space, $\tilde{O}(1)$ query
$T \subset V$	Naïve	$\tilde{O}(\|T\| \cdot k)$ space, $O(1)$ query

Results



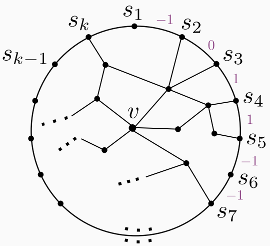
	Solution	Complexity
$T = S$	Unit-Monge [AGMW'18]	$\tilde{O}(k)$ space, $\tilde{O}(1)$ query
$T = V$	MSSP [Klein'05]	$O(n)$ space, $\tilde{O}(1)$ query
$T \subset V$	Naïve	$\tilde{O}(\|T\| \cdot k)$ space, $O(1)$ query
$T \subset V$	VC-dimension [LiParter'19]	$\tilde{O}(\|T\| + k^4)$ space, $O(1)$ query

Results



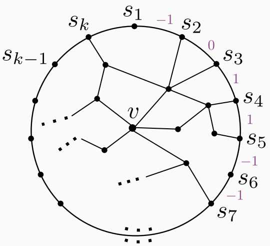
	Solution	Complexity
$T = S$	Unit-Monge [AGMW'18]	$\tilde{O}(k)$ space, $\tilde{O}(1)$ query
$T = V$	MSSP [Klein'05]	$O(n)$ space, $\tilde{O}(1)$ query
$T \subset V$	Naïve	$\tilde{O}(\|T\| \cdot k)$ space, $O(1)$ query
$T \subset V$	VC-dimension [LiParter'19]	$\tilde{O}(\|T\| + k^4)$ space, $O(1)$ query
$T \subset V$	**Today**	$\tilde{O}(\|T\| + k^3)$ space, $\tilde{O}(1)$ query

The Pattern of $v \in V$



$$p_v = \langle d(v,s_2) - d(v,s_1), d(v,s_3) - d(v,s_2), \cdots, d(v,s_k) - d(v,s_{k-1}) \rangle$$

The Pattern of $v \in V$

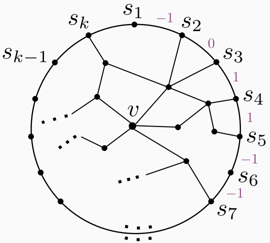


$$p_v = \langle d(v, s_2) - d(v, s_1), d(v, s_3) - d(v, s_2), \cdots, d(v, s_k) - d(v, s_{k-1}) \rangle$$

- $p_v \in \{-1, 0, 1\}^{k-1}$ by the triangle inequality.
- v-to-s_i distances are determined by p_v and $d(v, s_1)$:

$$d(v, s_i) = d(v, s_1) + \underbrace{\sum_{j=1}^{i-1} p_v[j]}_{\text{prefix-sum}}$$

The Pattern of $v \in V$



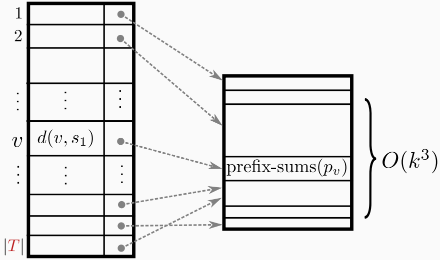
Theorem (Li & Parter [STOC 2019])

There are only $p_{\#} = O(k^3)$ distinct patterns among all vertices of the graph.

Huge improvement over the trivial $O(3^k)$ bound. Proved using a VC-dimension argument.

Li & Parter's Compression

1. One table with the $O(k^3)$ distinct patterns and their prefix-sums.
2. Every $v \in T$ stores $d(v, s_1)$ and a pointer to p_v in the right table.



Space: $\tilde{O}(|T| + k^4)$, Query time: $O(1)$.

Our Algorithmic Results

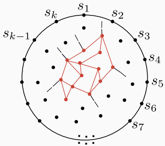
Let $p_{\#}$ = the number of distinct patterns among all vertices of G.

- An $\tilde{O}(|T| + p_{\#} + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.

Our Algorithmic Results

Let $p_{\#}$ = the number of distinct patterns among all vertices of G.

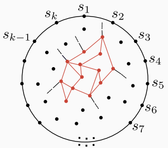
- An $\tilde{O}(|T| + p_{\#} + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.
- An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a *connected* subgraph of G,



Our Algorithmic Results

Let $p_{\#}$ = the number of distinct patterns among all vertices of G.

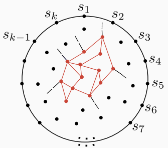
- An $\tilde{O}(|T| + p_{\#} + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.
- An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a *connected* subgraph of G,



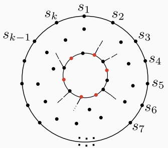
Our Algorithmic Results

Let $p_\#$ = the number of distinct patterns among all vertices of G.

- An $\tilde{O}(|T| + p_\# + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.
- An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a *connected* subgraph of G,



  and (2) T lies on a single face (not necessarily consecutively).

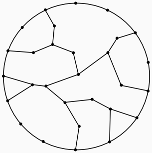


Our Combinatorial Results

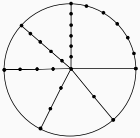
- An alternative $p_{\#} = O(k^3)$ proof that exploits planarity beyond VC-dimension. Namely, planar duality and the fact that distances among vertices of S are bounded by k.

Our Combinatorial Results

- An alternative $p_{\#} = O(k^3)$ proof that exploits planarity beyond VC-dimension. Namely, planar duality and the fact that distances among vertices of S are bounded by k.
- In Halin graphs, we show that $p_{\#} = \Theta(k^2)$, whereas the VC-dimension argument is limited to showing $O(k^3)$.



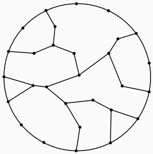
Tree $\cup$ Cycle



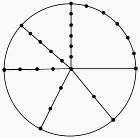
$\Omega(k^2)$ patterns

Our Combinatorial Results

- An alternative $p_{\#} = O(k^3)$ proof that exploits planarity beyond VC-dimension. Namely, planar duality and the fact that distances among vertices of S are bounded by k.
- In Halin graphs, we show that $p_{\#} = \Theta(k^2)$, whereas the VC-dimension argument is limited to showing $O(k^3)$.



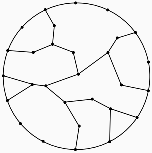
Tree $\cup$ Cycle



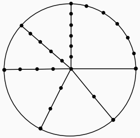
$\Omega(k^2)$ patterns

Our Combinatorial Results

- An alternative $p_{\#} = O(k^3)$ proof that exploits planarity beyond VC-dimension. Namely, planar duality and the fact that distances among vertices of S are bounded by k.
- In Halin graphs, we show that $p_{\#} = \Theta(k^2)$, whereas the VC-dimension argument is limited to showing $O(k^3)$.



Tree $\cup$ Cycle



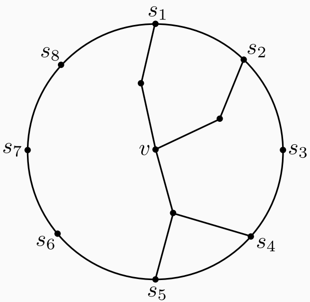
$\Omega(k^2)$ patterns

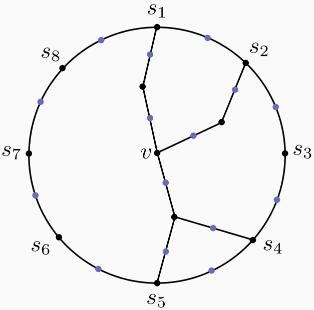
Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

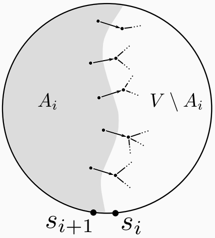
Simplifying Assumption: Binary Patterns

Assume w.l.o.g. that the patterns are over $\{-1, 1\}$ and not $\{-1, 0, 1\}$.
This can be achieved by subdividing every edge:





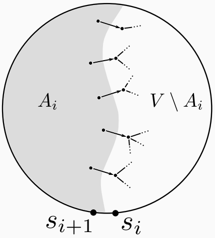
Our Framework: Bisectors



For every $1 \leq i < k$, define the following cuts:

$$A_i = \{v \in V \mid p_v[i] = -1\}$$
$$V \setminus A_i = \{v \in V \mid p_v[i] = 1\}$$

Our Framework: Bisectors

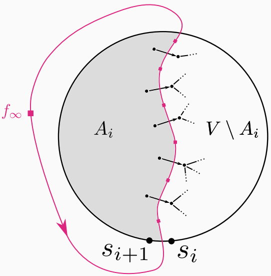


For every $1 \leq i < k$, define the following cuts:

$$A_i = \{v \in V \mid p_v[i] = -1\}$$
$$V \setminus A_i = \{v \in V \mid p_v[i] = 1\}$$

A *bisector* is the set of dual arcs $\beta_i := \{(uv)^* \mid u \in A_i\,,\ v \in V \setminus A_i\}$.

Our Framework: Bisectors



For every $1 \leq i < k$, define the following cuts:

$$A_i = \{v \in V \mid p_v[i] = -1\}$$
$$V \setminus A_i = \{v \in V \mid p_v[i] = 1\}$$

A *bisector* is the set of dual arcs $\beta_i := \{(uv)^* \mid u \in A_i\,,\ v \in V \setminus A_i\}$.

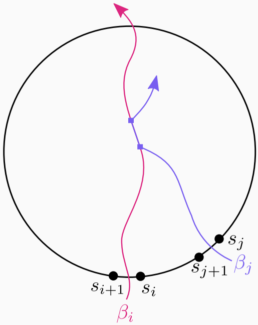
β_i **is a simple cycle in the dual graph (oriented counter-clockwise)**

Our Algorithmic Results

Let $p_{\#}$ = the number of distinct patterns among all vertices of G.

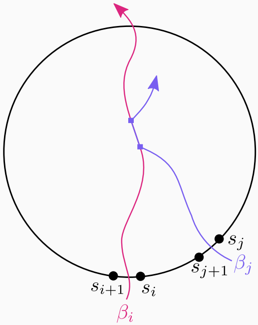
- **An $\tilde{O}(|T| + p_{\#} + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.**
- An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a *connected* subgraph of G, and (2) T lies on a single face (not necessarily consecutively).

Every Two Bisectors are Arc-disjoint



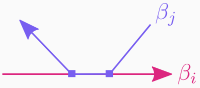
This is a forbidden configuration!

Every Two Bisectors are Arc-disjoint



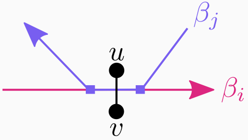
This is a forbidden configuration!

However, it is possible that β_i contains *reversed* arcs of β_j:



Patterns of Adjacent Vertices Differ in at Most Two Bits

For any $\{u, v\} \in E(G)$, u and v are separated by at most two bisectors.

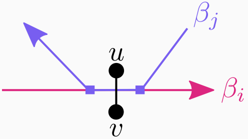


$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots - 1, -1 \rangle$$

$$\langle 1, -1, \cdots, 1, \cdots, -1, \cdots - 1, -1 \rangle$$

Patterns of Adjacent Vertices Differ in at Most Two Bits

For any $\{u, v\} \in E(G)$, u and v are separated by at most two bisectors.

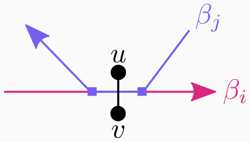


$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots -1, -1 \rangle$$
$$\langle 1, -1, \cdots, 1, \cdots, -1, \cdots -1, -1 \rangle$$

Hence, p_u and p_v differ in at most two bits.

Patterns of Adjacent Vertices Differ in at Most Two Bits

For any $\{u, v\} \in E(G)$, u and v are separated by at most two bisectors.



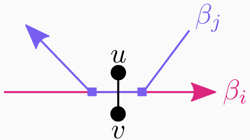
$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots - 1, -1 \rangle$$
$$\langle 1, -1, \cdots, 1, \cdots, -1, \cdots - 1, -1 \rangle$$

Hence, p_u and p_v differ in at most two bits.

We can use this fact to get a compression of size $\tilde{O}(|T| + p_\# + k)$.

Patterns of Adjacent Vertices Differ in at Most Two Bits

For any $\{u, v\} \in E(G)$, u and v are separated by at most two bisectors.



$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots - 1, -1 \rangle$$
$$\langle 1, -1, \cdots, 1, \cdots, -1, \cdots - 1, -1 \rangle$$

Hence, p_u and p_v differ in at most two bits.

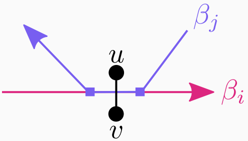
We can use this fact to get a compression of size $\tilde{O}(|T| + p_{\#} + k)$.

Idea:

1. Store one pattern explicitly using $O(k)$ bits.

Patterns of Adjacent Vertices Differ in at Most Two Bits

For any $\{u, v\} \in E(G)$, u and v are separated by at most two bisectors.



$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots - 1, -1 \rangle$$
$$\langle 1, -1, \cdots, 1, \cdots, -1, \cdots - 1, -1 \rangle$$

Hence, p_u and p_v differ in at most two bits.

We can use this fact to get a compression of size $\tilde{O}(|T| + p_{\#} + k)$.

Idea:

1. Store one pattern explicitly using $O(k)$ bits.
2. Traverse the graph and record the $O(p_{\#})$ changes between different patterns using a persistent data structure for prefix-sums.

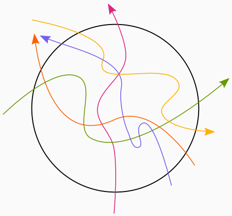
Our Combinatorial Results

- **An alternative $p_{\#} = O(k^3)$ proof that exploits planarity beyond VC-dimension. Namely, planar duality and the fact that distances among vertices of S are bounded by k.**
- In Halin graphs, we show that $p_{\#} = \Theta(k^2)$, whereas the VC-dimension argument is limited to showing $O(k^3)$.

The Pattern Graph $G_{\mathcal{P}}$

Definition

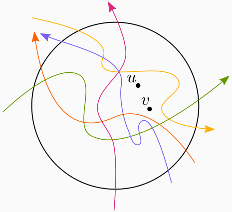
The *Pattern graph* $G_{\mathcal{P}}$ is composed of the union of all the bisectors.



The Pattern Graph $G_{\mathcal{P}}$

Definition

The *Pattern graph* $G_{\mathcal{P}}$ is composed of the union of all the bisectors.

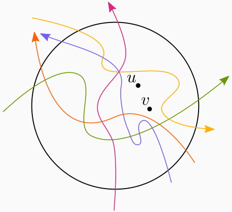


Every $u, v \in V$ embedded in the same face have the same pattern.

The Pattern Graph $G_{\mathcal{P}}$

Definition

The *Pattern graph* $G_{\mathcal{P}}$ is composed of the union of all the bisectors.



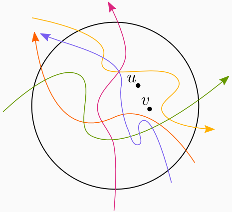
Every $u, v \in V$ embedded in the same face have the same pattern.

Idea: bound the number of faces of $G_{\mathcal{P}}$.

The Pattern Graph $G_{\mathcal{P}}$

Definition

The *Pattern graph* $G_{\mathcal{P}}$ is composed of the union of all the bisectors.



Every $u, v \in V$ embedded in the same face have the same pattern.

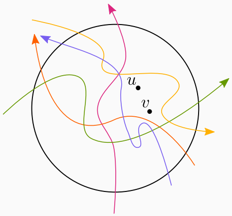
Idea: bound the number of faces of $G_{\mathcal{P}}$.

By Euler's formula, the number of faces in $G_{\mathcal{P}}$ is bounded by the number of *crossings*.

The Pattern Graph $G_{\mathcal{P}}$

Definition

The *Pattern graph* $G_{\mathcal{P}}$ is composed of the union of all the bisectors.



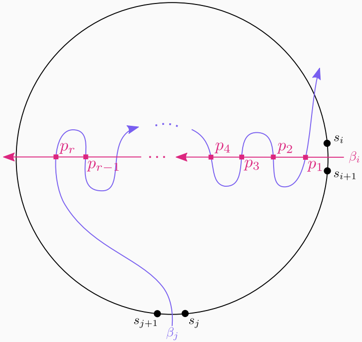
Every $u, v \in V$ embedded in the same face have the same pattern.

Idea: bound the number of faces of $G_{\mathcal{P}}$.

By Euler's formula, the number of faces in $G_{\mathcal{P}}$ is bounded by the number of *crossings*.

Our main technical contribution: every two bisectors can cross at most $O(k)$ times, hence the number of crossings is $O(k^3)$.

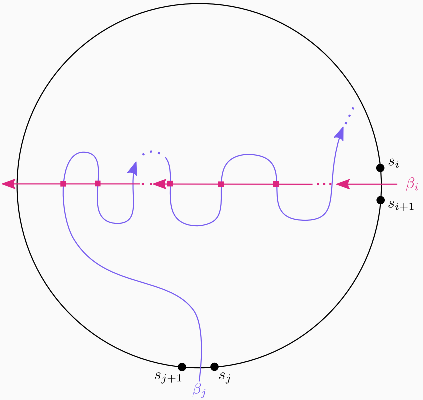
Every Two Bisectors Cross in Opposite Orientation



Lemma

Let $p_1, p_2, \ldots, p_r$ be the crossing points of β_i and β_j, in the order they appear along β_i.
The crossing points along β_j are reversed $p_r, p_{r-1}, \ldots, p_1$.

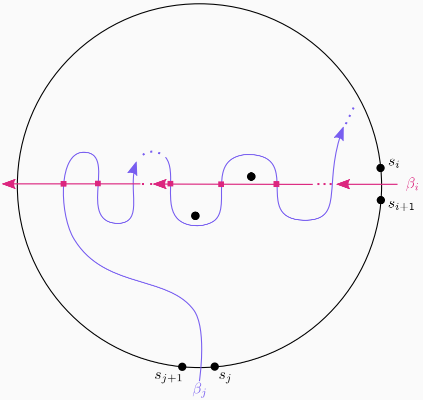
Two Bisectors can Cross at Most $O(k)$ Times



Lemma

The number of crossings between β_i and β_j is $r = O(k)$.

Two Bisectors can Cross at Most $O(k)$ Times

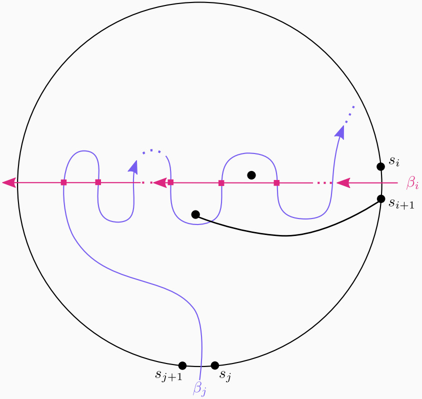


Lemma

The number of crossings between β_i and β_j is $r = O(k)$.

Idea: Look at the shortest paths from the vertices inside the "pockets".

Two Bisectors can Cross at Most $O(k)$ Times

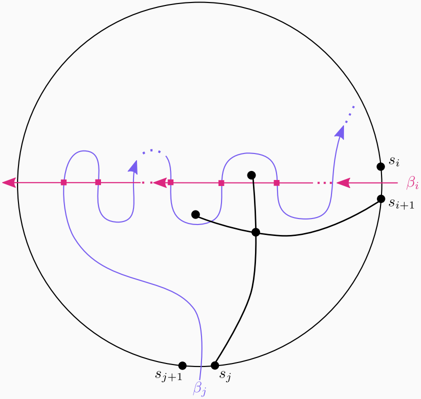


Lemma

The number of crossings between β_i and β_j is $r = O(k)$.

Idea: Look at the shortest paths from the vertices inside the "pockets".

Two Bisectors can Cross at Most $O(k)$ Times

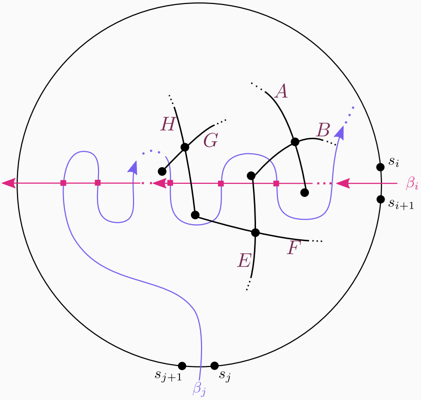


Lemma

The number of crossings between β_i and β_j is $r = O(k)$.

Idea: Look at the shortest paths from the vertices inside the "pockets".

Two Bisectors can Cross at Most $O(k)$ Times

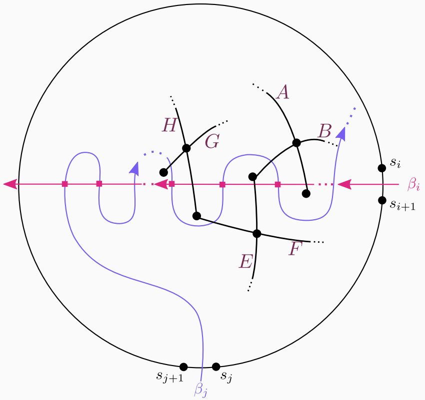


Lemma

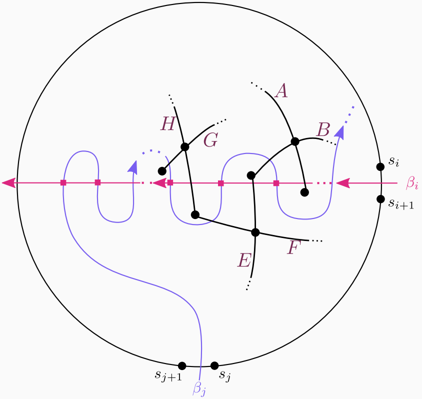
The number of crossings between β_i and β_j is $r = O(k)$.

Idea: Look at the shortest paths from the vertices inside the "pockets".

Two Bisectors can Cross at Most $O(k)$ Times

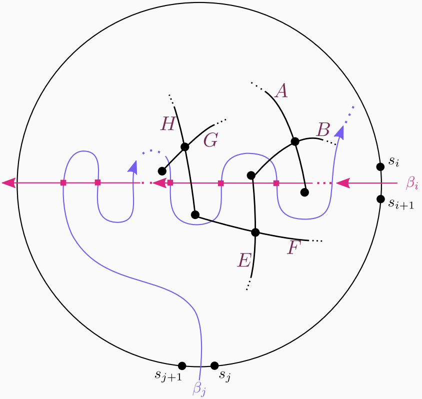


Two Bisectors can Cross at Most $O(k)$ Times



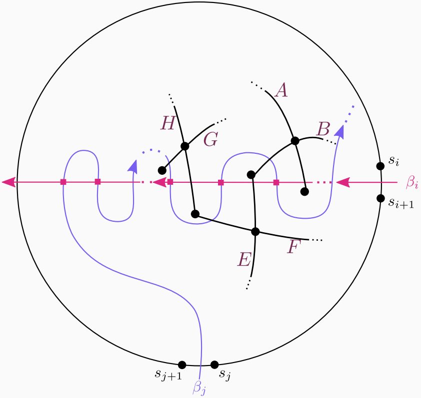
We get: $A - B \geq E - F + 2$

Two Bisectors can Cross at Most $O(k)$ Times



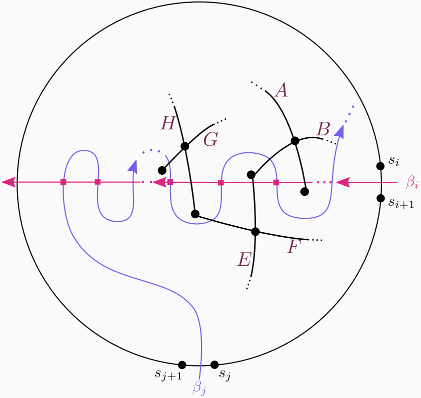
We get: $A - B \geq E - F + 2 \geq H - G + 4$

Two Bisectors can Cross at Most $O(k)$ Times



We get: $A - B \geq E - F + 2 \geq H - G + 4 \geq \ldots \geq \Omega(r)$

Two Bisectors can Cross at Most $O(k)$ Times



We get: $A - B \geq E - F + 2 \geq H - G + 4 \geq \ldots \geq \Omega(r)$

Hence:

$$\boldsymbol{\Omega(r) \leq A - B \leq k}$$

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

2. We've shown how to compress the $T \times S$ distances into $\tilde{O}(|T| + p_{\#} + k)$ bits.

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

2. We've shown how to compress the $T \times S$ distances into $\tilde{O}(|T| + p_{\#} + k)$ bits.
    - A simple argument shows that $\Omega(|T| + k)$ bits are necessary. Are the additive $\tilde{O}(p_{\#})$ bits in the construction necessary?

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

2. We've shown how to compress the $T \times S$ distances into $\tilde{O}(|T| + p_{\#} + k)$ bits.
    - A simple argument shows that $\Omega(|T| + k)$ bits are necessary. Are the additive $\tilde{O}(p_{\#})$ bits in the construction necessary?
    - For small values of $|T|$, the naïve compression of $\tilde{O}(|T| \cdot k)$ bits is still the best known. Can it be improved?

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

2. We've shown how to compress the $T \times S$ distances into $\tilde{O}(|T| + p_{\#} + k)$ bits.
   - A simple argument shows that $\Omega(|T| + k)$ bits are necessary. Are the additive $\tilde{O}(p_{\#})$ bits in the construction necessary?
   - For small values of $|T|$, the naïve compression of $\tilde{O}(|T| \cdot k)$ bits is still the best known. Can it be improved?

Open Questions

1. There is a remaining gap between our example that achieves $\Theta(k^2)$ patterns, and the $O(k^3)$ upper bound.

Conjecture

The number of distinct patterns over all vertices of a planar graph is $O(k^2)$.

2. We've shown how to compress the $T \times S$ distances into $\tilde{O}(|T| + p_\# + k)$ bits.
   - A simple argument shows that $\Omega(|T| + k)$ bits are necessary. Are the additive $\tilde{O}(p_\#)$ bits in the construction necessary?
   - For small values of $|T|$, the naïve compression of $\tilde{O}(|T| \cdot k)$ bits is still the best known. Can it be improved?

The End