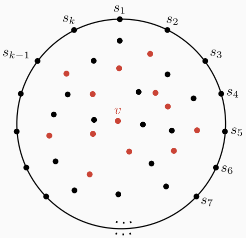
Improved Compression of the Okamura-Seymour Metric

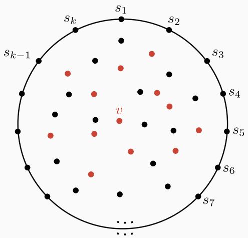
Shay Mozes, Nathan Wallheimer, Oren Weimann

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

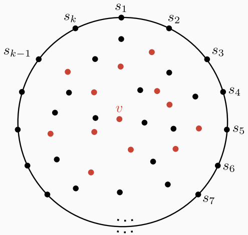


- An undirected, unweighted planar graph $G = (V, E)$.
- A set $S = \{s_1, s_2, \ldots, s_k\}$ of k consecutive vertices on a face f_∞.
- A set $T \subseteq V$ of terminal vertices lying anywhere in the graph.

The Okamura-Seymour Metric Compression Problem

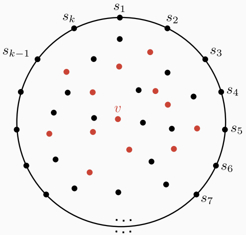
Succinctly encode the $T \times S$ distances to answer $d(v, s_i)$ queries.

Results



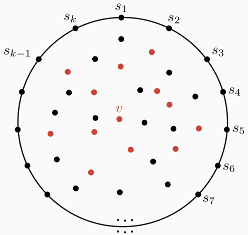
	Solution	Complexity
$T = S$	Unit-Monge [AGMW'18]	$\tilde{O}(k)$ space, $\tilde{O}(1)$ query
$T = V$	MSSP [Klein'05]	$O(n)$ space, $\tilde{O}(1)$ query
$T \subset V$	Naïve	$\tilde{O}(\lvert T \rvert \cdot k)$ space, $O(1)$ query

Results



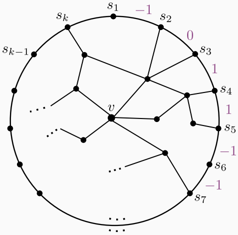
	Solution	Complexity
$T = S$	Unit-Monge [AGMW'18]	$\tilde{O}(k)$ space, $\tilde{O}(1)$ query
$T = V$	MSSP [Klein'05]	$O(n)$ space, $\tilde{O}(1)$ query
$T \subset V$	Naïve	$\tilde{O}(\lvert T \rvert \cdot k)$ space, $O(1)$ query
$T \subset V$	VC-dimension [LiParter'19]	$\tilde{O}(\lvert T \rvert + k^4)$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

Results



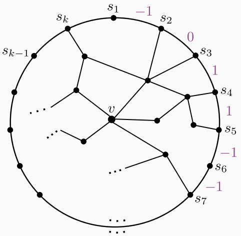
	Solution	Complexity
$T = S$	Unit-Monge [AGMW'18]	$\tilde{O}(k)$ space, $\tilde{O}(1)$ query
$T = V$	MSSP [Klein'05]	$O(n)$ space, $\tilde{O}(1)$ query
$T \subset V$	Naïve	$\tilde{O}(\lvert T\rvert \cdot k)$ space, $O(1)$ query
$T \subset V$	VC-dimension [LiParter'19]	$\tilde{O}(\lvert T\rvert + k^4)$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

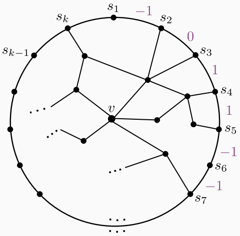


$$p_v = \langle d(v, s_2) - d(v, s_1), d(v, s_3) - d(v, s_2), \cdots, d(v, s_k) - d(v, s_{k-1}) \rangle$$

- $p_v \in \{-1, 0, 1\}^{k-1}$ 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 $v \in V$



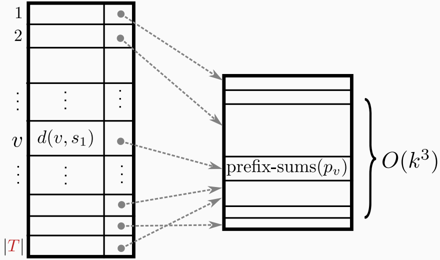
Theorem (Li & Parter [STOC 2019])

There are only $x = O(k^3)$ distinct patterns among all vertices of the graph.

Huge improvement over the trivial $O(3^k)$ bound.

Li & Parter's Compression

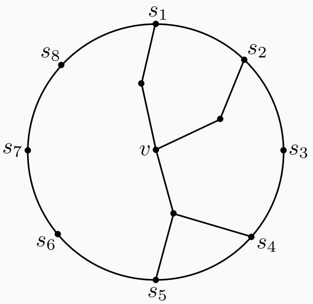
1. One table with the $O(k^3)$ distinct patterns and their prefix-sums.
2. Every $v \in T$ stores $d(v, s_1)$ and a pointer to p_v in the previous table.



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

Li & Parter's $O(k^3)$ Proof

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:



Li & Parter's $O(k^3)$ Proof

Arrange the n binary patterns as the rows of a binary matrix.
By planarity, there are no four columns $a < b < c < d$ such that for some $u, v \in V$:

$$\begin{array}{c} \\ \vdots \\ u \\ \vdots \\ v \\ \vdots \end{array} \begin{array}{c} \begin{matrix} a & b & c & d \end{matrix} \\ \begin{pmatrix} & \vdots & \vdots & \\ -1 & 1 & -1 & 1 \\ & \vdots & \vdots & \\ 1 & -1 & 1 & -1 \\ & \vdots & \vdots & \end{pmatrix} \end{array}$$

Li & Parter's $O(k^3)$ Proof

Arrange the n binary patterns as the rows of a binary matrix.
By planarity, there are no four columns $a < b < c < d$ such that for some $u, v \in V$:

$$\begin{array}{c} \\ \vdots \\ u \\ \vdots \\ v \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} & \vdots & \vdots & \\ -1 & 1 & -1 & 1 \\ & \vdots & \vdots & \\ 1 & -1 & 1 & -1 \\ & \vdots & \vdots & \end{pmatrix} \end{array}$$

Thus, the VC-dimension of the matrix is at most 3.
By the Sauer-Shelach Lemma, the number of distinct rows is $O(k^3)$.

Our Results

Let $x =$ the number of distinct patterns among all vertices of G.

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

Our Results

Let x = the number of distinct patterns among all vertices of G.

- An $\tilde{O}(|T| + x + k)$ 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 Results

Let $x =$ the number of distinct patterns among all vertices of G.

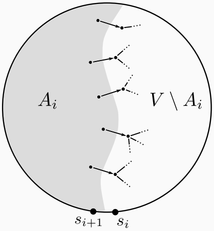
- An $\tilde{O}(|T| + x + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.
- An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a *connected* subgraph of G, and (2) T lies on a single face (not necessarily consecutively).
- An alternative $x = O(k^3)$ proof that exploits planarity beyond VC-dimension. Namely, planar duality and the fact that distances among vertices of S are bounded by k.

Our Results

Let x = the number of distinct patterns among all vertices of G.

- An $\tilde{O}(|T| + x + k)$ 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).
- An alternative $x = O(k^3)$ 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 $x = \Theta(k^2)$ while the VC-dimension argument is limited to showing $O(k^3)$.

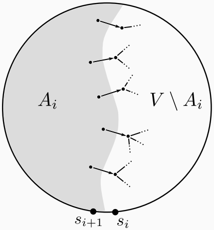
Bisectors



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

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

Bisectors

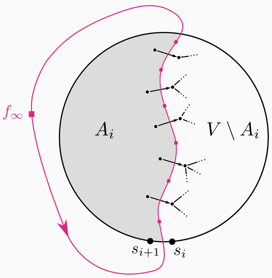


For every $1 \leq i < k$, define the following cuts:

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

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

Bisectors

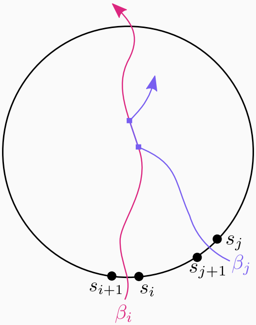


For every $1 \leq i < k$, define the following cuts:

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

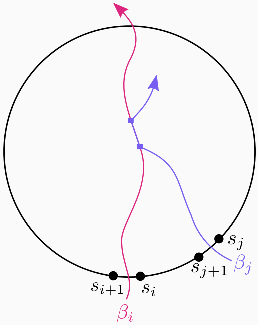
A *bisector* is the set of dual arcs $\beta_i := \{(uv)^* \mid u \in A_i\,,\ v \in V \setminus A_i\}$.
β_i **is a directed simple cycle in the dual graph.**

Every Two Bisectors are Arc-Disjoint



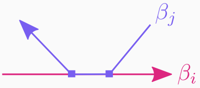
This is a forbidden configuration!

Every Two Bisectors are Arc-Disjoint



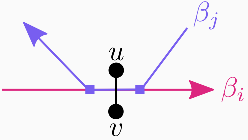
This is a forbidden configuration!

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.

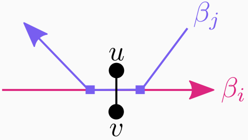


$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots - 1, -1 \rangle$$

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

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.

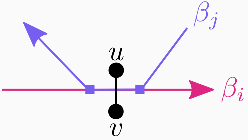


$$\langle 1, -1, \cdots, -1, \cdots, 1, \cdots - 1, -1 \rangle$$
$$\langle 1, -1, \cdots, 1, \cdots, -1, \cdots - 1, -1 \rangle$$

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.

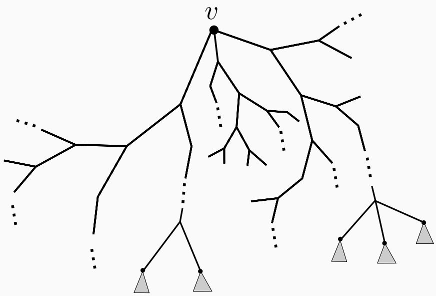
We can use this fact to get a compression of size $\tilde{O}(|T| + x + k)$

Our Results

Let x = the number of distinct patterns among all vertices of G.

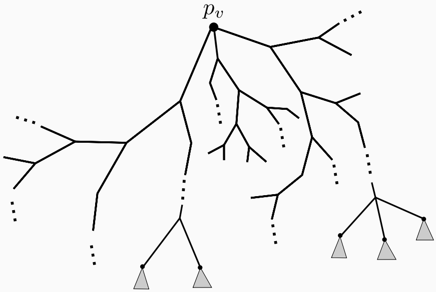
- **An $\tilde{O}(|T| + x + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.**
- An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a *connected* subgraph of G, and (2) T lies on a single face.
- An alternative $x = O(k^3)$ 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 $x = \Theta(k^2)$ while the VC-dimension argument is limited to showing $O(k^3)$.

A Spanning Tree of G



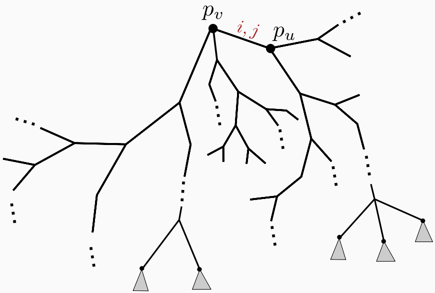
- Root the tree at an arbitrary vertex v.

A Spanning Tree of G



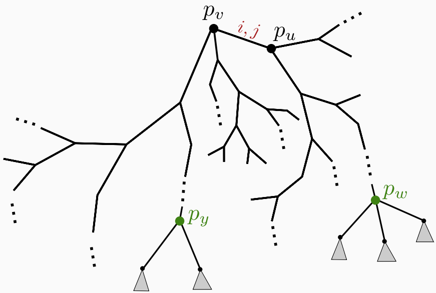
- Root the tree at an arbitrary vertex v.
- Identify each node with its pattern.

A Spanning Tree of G



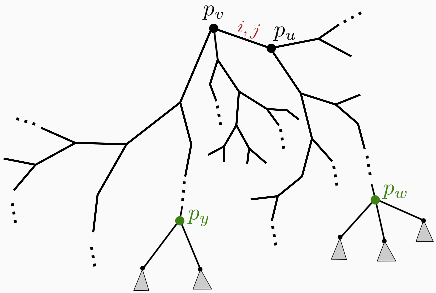
- Root the tree at an arbitrary vertex v.
- Identify each node with its pattern.
- Label every edge by the two bits that change.

Some Patterns Appear Multiple Times



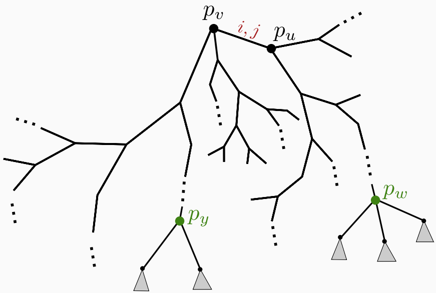
1. Let y and w be two vertices such that $p_y = p_w$.

Some Patterns Appear Multiple Times



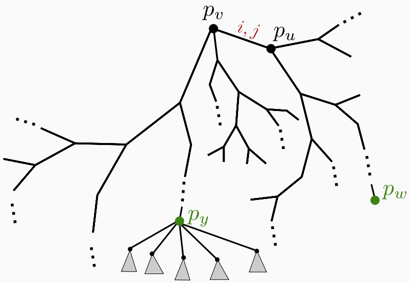
1. Let y and w be two vertices such that $p_y = p_w$.
2. Assume w.l.o.g. that $depth(y) \leq depth(w)$.

Some Patterns Appear Multiple Times



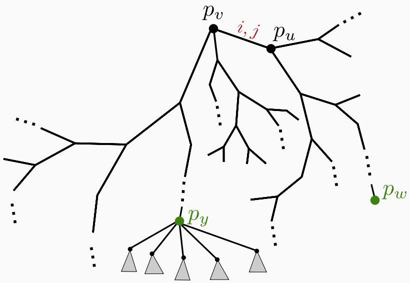
1. Let y and w be two vertices such that $p_y = p_w$.
2. Assume w.l.o.g. that $depth(y) \leq depth(w)$.
3. Move w's children to y (edge labels remain the same).

Some Patterns Appear Multiple Times



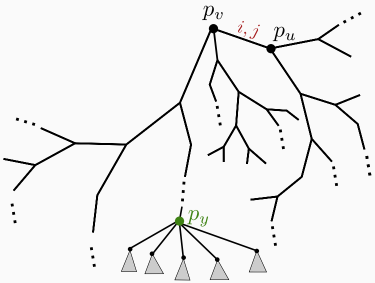
1. Let y and w be two vertices such that $p_y = p_w$.
2. Assume w.l.o.g. that $depth(y) \leq depth(w)$.
3. Move w's children to y (edge labels remain the same).

Some Patterns Appear Multiple Times



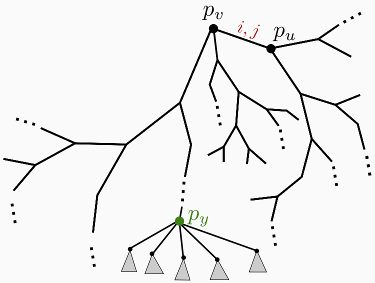
1. Let y and w be two vertices such that $p_y = p_w$.
2. Assume w.l.o.g. that $depth(y) \leq depth(w)$.
3. Move w's children to y (edge labels remain the same).
4. Remove w from the tree.

Some Patterns Appear Multiple Times



1. Let y and w be two vertices such that $p_y = p_w$.
2. Assume w.l.o.g. that $depth(y) \leq depth(w)$.
3. Move w's children to y (edge labels remain the same).
4. Remove w from the tree.

Some Patterns Appear Multiple Times



1. Let y and w be two vertices such that $p_y = p_w$.
2. Assume w.l.o.g. that $depth(y) \leq depth(w)$.
3. Move w's children to y (edge labels remain the same).
4. Remove w from the tree.
5. Repeat until the size of the tree is x.

The Data Structure

The Data Structure

1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.

The Data Structure

1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.
2. As we *traverse the tree*, only $O(x)$ bit-changes occur.

The Data Structure

1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.
2. As we *traverse the tree*, only $O(x)$ bit-changes occur.
3. Maintain all of those $O(x)$ versions of the pattern in a *persistent* data structure for prefix-sum that supports updates.

The Data Structure

1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.
2. As we *traverse the tree*, only $O(x)$ bit-changes occur.
3. Maintain all of those $O(x)$ versions of the pattern in a *persistent* data structure for prefix-sum that supports updates.
   Space: $\tilde{O}(x + k)$ and $\tilde{O}(1)$ query.

The Data Structure

1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.
2. As we *traverse the tree*, only $O(x)$ bit-changes occur.
3. Maintain all of those $O(x)$ versions of the pattern in a *persistent* data structure for prefix-sum that supports updates.
   Space: $\tilde{O}(x + k)$ and $\tilde{O}(1)$ query.
4. Every $v \in T$ stores $d(v, s_1)$ and a pointer to the version of the prefix-sum data structure at p_v.

The Data Structure

1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.
2. As we *traverse the tree*, only $O(x)$ bit-changes occur.
3. Maintain all of those $O(x)$ versions of the pattern in a *persistent* data structure for prefix-sum that supports updates.
   Space: $\tilde{O}(x + k)$ and $\tilde{O}(1)$ query.
4. Every $v \in T$ stores $d(v, s_1)$ and a pointer to the version of the prefix-sum data structure at p_v.

In total: $\tilde{O}(|T| + x + k)$ space and $\tilde{O}(1)$ time for query.

The Data Structure

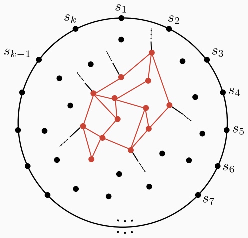
1. Construct a tree of size x whose nodes correspond to patterns, and edges are labeled by the bits that change.
2. As we *traverse the tree*, only $O(x)$ bit-changes occur.
3. Maintain all of those $O(x)$ versions of the pattern in a *persistent* data structure for prefix-sum that supports updates.
   Space: $\tilde{O}(x + k)$ and $\tilde{O}(1)$ query.
4. Every $v \in T$ stores $d(v, s_1)$ and a pointer to the version of the prefix-sum data structure at p_v.

In total: $\tilde{O}(|T| + x + k)$ space and $\tilde{O}(1)$ time for query.
Preprocessing: can be done in $\tilde{O}(n)$ time.

Let $x =$ the number of distinct patterns among all vertices of G.

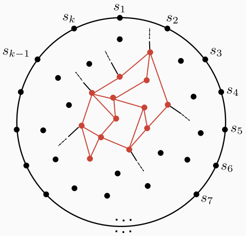
- An $\tilde{O}(|T| + x + k)$ bits compression of the Okamura-Seymour metric, with query time $\tilde{O}(1)$.
- **An optimal $\tilde{O}(|T| + k)$ bits compression with $\tilde{O}(1)$ query for two special cases: (1) T induces a connected subgraph of G, and (2) T lies on a single face.**
- An alternative $x = O(k^3)$ 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 $x = \Theta(k^2)$ while the VC-dimension argument is limited to showing $O(k^3)$.

T Induces a Connected Component



We only need to traverse T. Thus, there are only $x = O(|T|)$ versions.

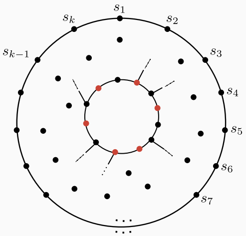
T Induces a Connected Component



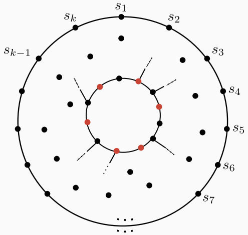
We only need to traverse T. Thus, there are only $x = O(|T|)$ versions.

Space: $\tilde{O}(|T| + k)$. Time: $\tilde{O}(1)$.

T Lies on a Single Face

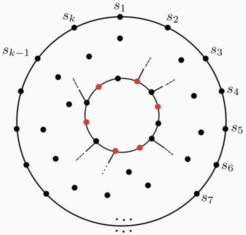


T Lies on a Single Face



We only need to traverse the vertices along the face. Every bisector can visit the face at most once. Thus, there are only $x = O(k)$ versions.

T Lies on a Single Face



We only need to traverse the vertices along the face. Every bisector can visit the face at most once. Thus, there are only $x = O(k)$ versions.

Space: $\tilde{O}(|T| + k)$. Time: $\tilde{O}(1)$.

Our Results

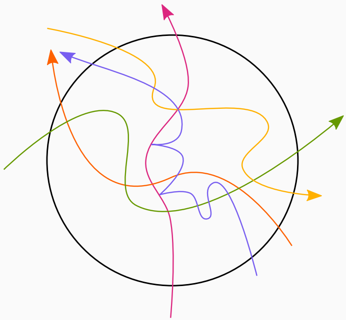
Let x = the number of distinct patterns among all vertices of G.

- An $\tilde{O}(|T| + x + k)$ 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.
- **An alternative $x = O(k^3)$ 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 $x = \Theta(k^2)$ while the VC-dimension argument is limited to showing $O(k^3)$.

The Bisector Graph $G_{\mathcal{B}}$

Definition

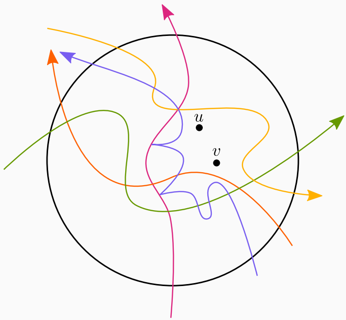
The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.



The Bisector Graph $G_{\mathcal{B}}$

Definition

The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.

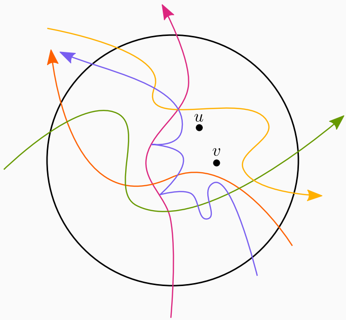


- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.

The Bisector Graph $G_{\mathcal{B}}$

Definition

The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.

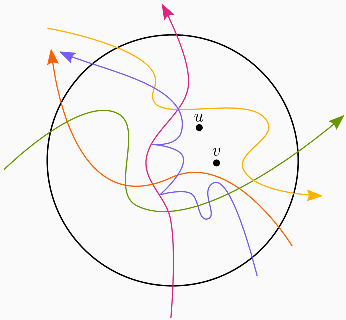


- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.
- If the number of faces in $G_{\mathcal{B}}$ was $O(k^3)$, we'd be done.

The Bisector Graph $G_{\mathcal{B}}$

Definition

The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.

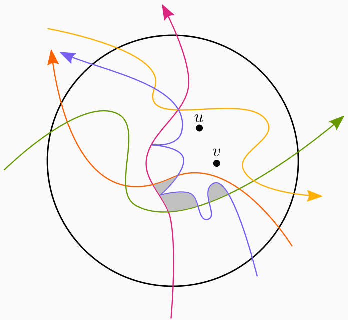


- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.
- If the number of faces in $G_{\mathcal{B}}$ was $O(k^3)$, we'd be done.
- Alas, there could be $\Omega(n)$ faces that correspond to the same pattern.

The Bisector Graph $G_{\mathcal{B}}$

Definition

The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.

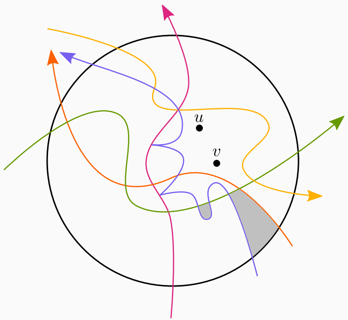


- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.
- If the number of faces in $G_{\mathcal{B}}$ was $O(k^3)$, we'd be done.
- Alas, there could be $\Omega(n)$ faces that correspond to the same pattern.

The Bisector Graph $G_{\mathcal{B}}$

Definition

The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.

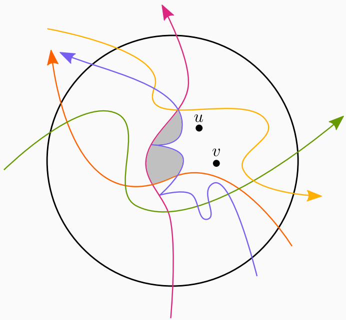


- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.
- If the number of faces in $G_{\mathcal{B}}$ was $O(k^3)$, we'd be done.
- Alas, there could be $\Omega(n)$ faces that correspond to the same pattern.

The Bisector Graph $G_{\mathcal{B}}$

Definition

The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.

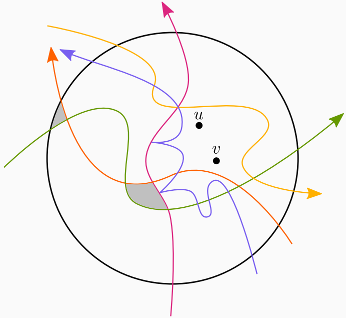


- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.
- If the number of faces in $G_{\mathcal{B}}$ was $O(k^3)$, we'd be done.
- Alas, there could be $\Omega(n)$ faces that correspond to the same pattern.

The Bisector Graph $G_{\mathcal{B}}$

Definition

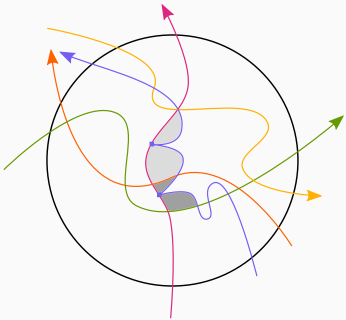
The *bisector graph* $G_{\mathcal{B}}$, is composed of the union of all the bisectors.



- For every $u, v \in V$ embedded in the same face, $p_u = p_v$.
- If the number of faces in $G_{\mathcal{B}}$ was $O(k^3)$, we'd be done.
- Alas, there could be $\Omega(n)$ faces that correspond to the same pattern.

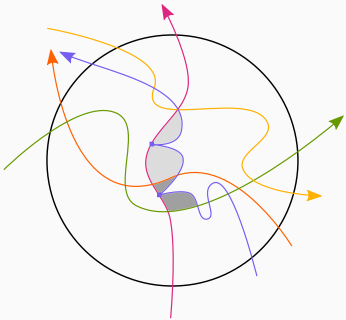
Remove Touching Points

A *touching* is an intersection of bisectors without *crossing*.



Remove Touching Points

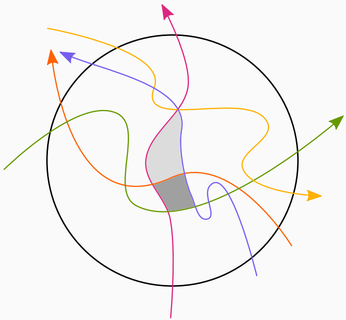
A *touching* is an intersection of bisectors without *crossing*.



Touching points can be removed such that vertices embedded in the same face will still have the same pattern.

Remove Touching Points

A *touching* is an intersection of bisectors without *crossing*.

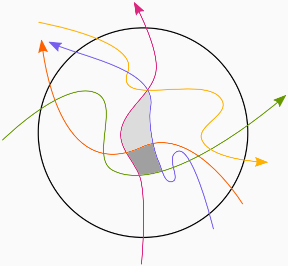


Touching points can be removed such that vertices embedded in the same face will still have the same pattern.

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

Definition

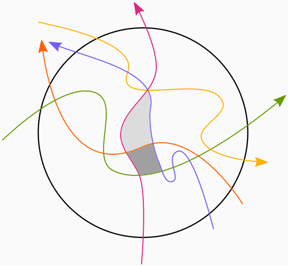
The *Pattern graph* $G_{\mathcal{P}}$ is obtained from $G_{\mathcal{B}}$ by removing all touching points.



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

Definition

The *Pattern graph* $G_{\mathcal{P}}$ is obtained from $G_{\mathcal{B}}$ by removing all touching points.

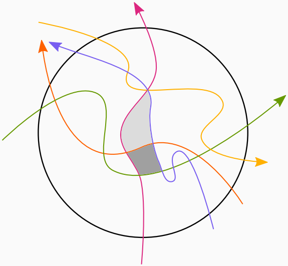


By Euler's formula, the number of faces in $\boldsymbol{G_{\mathcal{P}}}$ is bounded by the number of *crossings*.

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

Definition

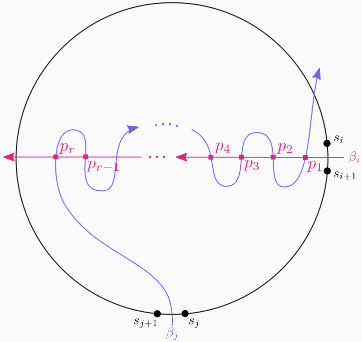
The *Pattern graph* $G_{\mathcal{P}}$ is obtained from $G_{\mathcal{B}}$ by removing all touching points.



By Euler's formula, the number of faces in $\boldsymbol{G_{\mathcal{P}}}$ 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 $O(k^3)$.

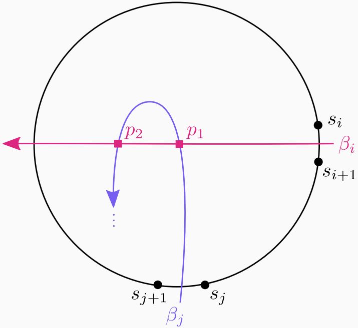
Every Two Bisectors Cross in Opposite Orientation



Lemma

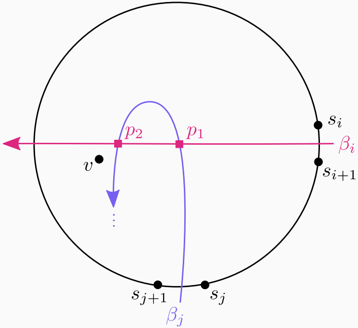
Let $p_1, p_2, \ldots, p_r$ be the crossing points of β_i and β_j, in the order they appear along β_i.
The crossing points along β_j are reversed $p_r, p_{r-1}, \ldots, p_1$.

Proof by Contradiction



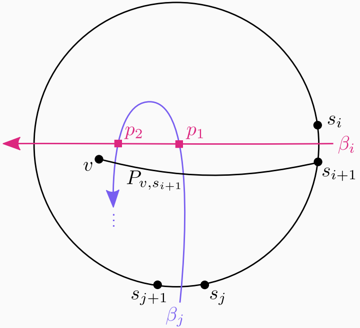
- Assume for contradition that p_1 appears before p_2 in β_j.

Proof by Contradiction



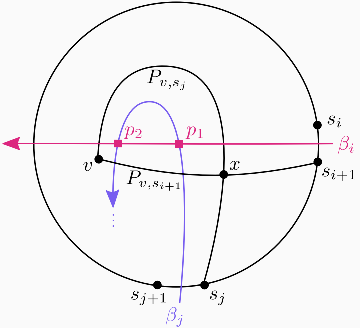
- Assume for contradition that p_1 appears before p_2 in β_j.
- Let $v \in V$ be the vertex that lies on the right of β_j after p_2.

Proof by Contradiction



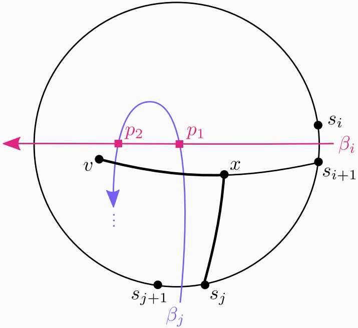
- Assume for contradition that p_1 appears before p_2 in β_j.
- Let $v \in V$ be the vertex that lies on the right of β_j after p_2.
- $P_{v,s_{i+1}}$ must stay on the left of β_i and cross β_j.

Proof by Contradiction



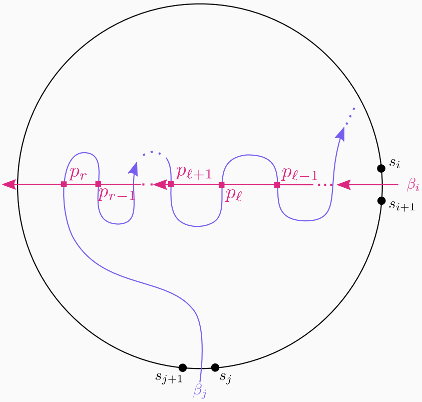
- Assume for contradition that p_1 appears before p_2 in β_j.
- Let $v \in V$ be the vertex that lies on the right of β_j after p_2.
- $P_{v,s_{i+1}}$ must stay on the left of β_i and cross β_j.
- P_{v,s_j} must stay on the right of β_j and cross $P_{v,s_{i+1}}$.

Proof by Contradiction



- Assume for contradiction that p_1 appears before p_2 in β_j.
- Let $v \in V$ be the vertex that lies on the right of β_j after p_2.
- $P_{v,s_{i+1}}$ must stay on the left of β_i and cross β_j.
- P_{v,s_j} must stay on the right of β_j and cross $P_{v,s_{i+1}}$.
- **There's a shortest v-to-s_j path that crosses β_j, a contradiction.**

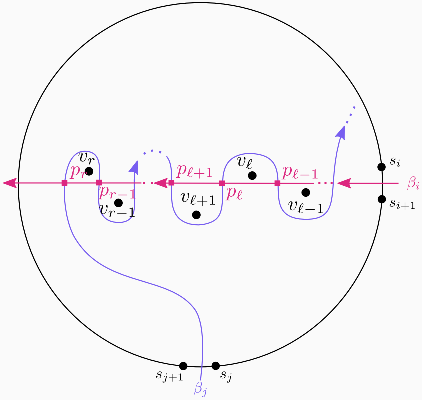
Two Bisectors can Cross at Most $O(k)$ Times



Lemma

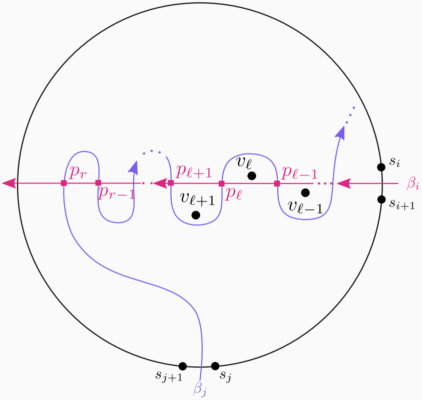
The number of crossings between β_i and β_j is $r = O(k)$.

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



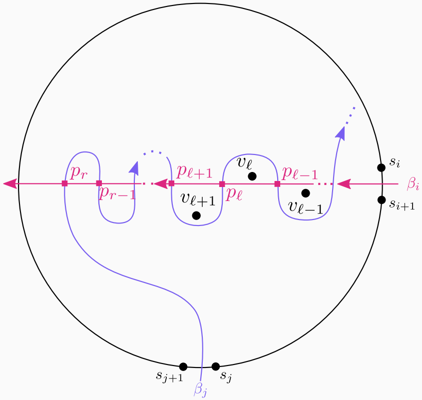
- Let $v_1, v_2, \ldots, v_r$ be primal vertices inside the "pockets" created between consecutive crossings.

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

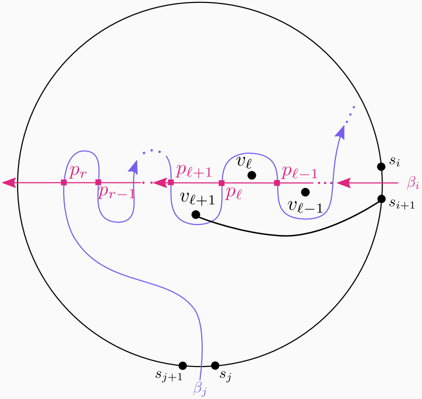


- Let $v_1, v_2, \ldots, v_r$ be primal vertices inside the "pockets" created between consecutive crossings.
- Consider some v_ℓ.

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

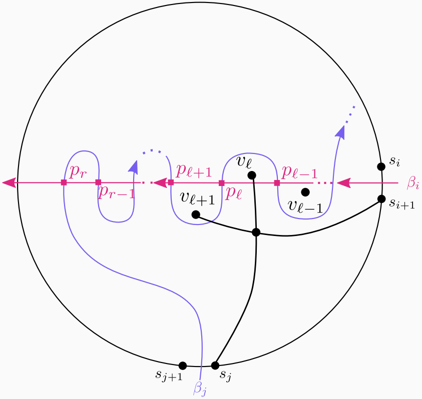


Two Bisectors can Cross at Most $O(k)$ Times



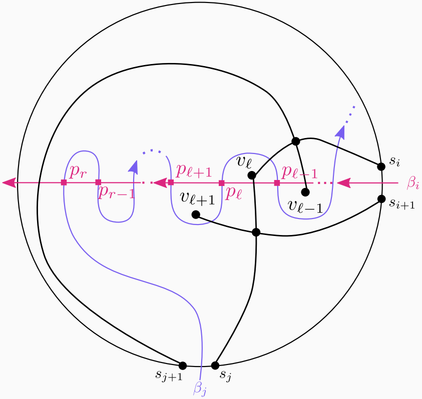
- $P_{v_{\ell+1},s_{i+1}}$ must remain on the left of β_i.

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



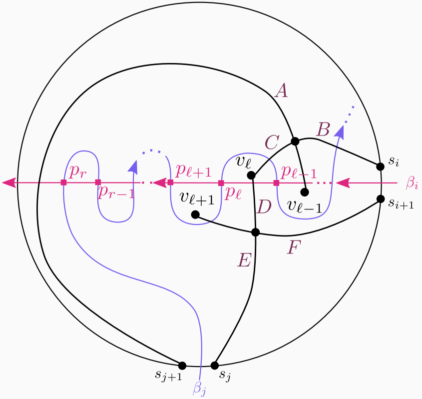
- $P_{v_{\ell+1},s_{i+1}}$ must remain on the left of β_i.
- P_{v_ℓ,s_j} must remain on the right of β_j and cross $P_{v_{\ell+1},s_{i+1}}$.

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



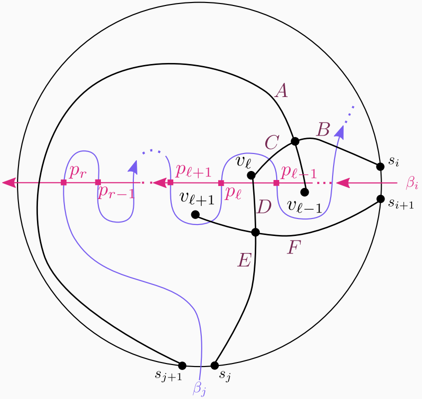
- $P_{v_{\ell+1},s_{i+1}}$ must remain on the left of β_i.
- P_{v_ℓ,s_j} must remain on the right of β_j and cross $P_{v_{\ell+1},s_{i+1}}$.
- A symmetric configuration with v_ℓ and $v_{\ell-1}$.

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



- $P_{v_{\ell+1},s_{i+1}}$ must remain on the left of β_i.
- P_{v_ℓ,s_j} must remain on the right of β_j and cross $P_{v_{\ell+1},s_{i+1}}$.
- A symmetric configuration with v_ℓ and $v_{\ell-1}$.
- Denote the lengths of subpaths by A, B, C, D, E, F.

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

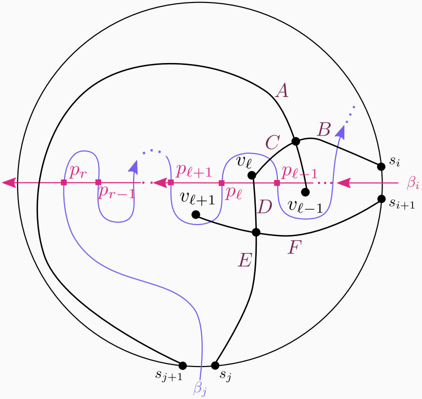


By the triangle inequality and since patterns are over $\{-1, 1\}$:

$$C + B \leq D + F - 1$$

$$D + E \leq C + A - 1$$

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



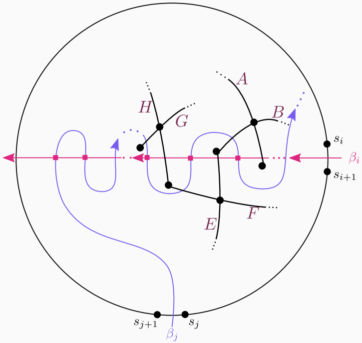
By the triangle inequality and since patterns are over $\{-1, 1\}$:

$$C + B \leq D + F - 1$$

$$D + E \leq C + A - 1$$

The sum of the inequalities is $E - F + 2 \leq A - B$.

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



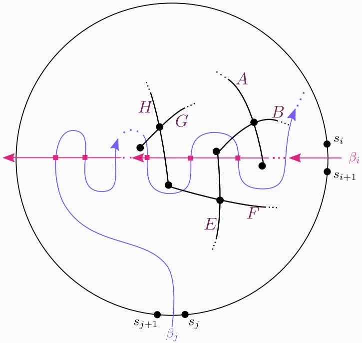
Symmetrically we get:

$$E - F + 2 \leq A - B$$

$$H - G + 2 \leq E - F$$

and so on...

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



Symmetrically we get:

$$E - F + 2 \leq A - B$$

$$H - G + 2 \leq E - F$$

and so on... **Hence, there exists a vertex *v* such that:**

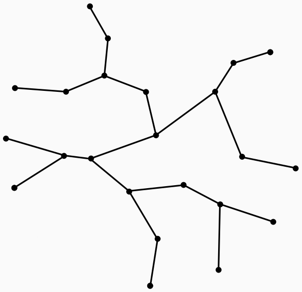
$$\Omega(\boldsymbol{r}) \leq \boldsymbol{d}(\boldsymbol{v}, \boldsymbol{s_i}) - \boldsymbol{d}(\boldsymbol{v}, \boldsymbol{s_{j+1}}) \leq \boldsymbol{k}$$

Our Results

Let x = the number of distinct patterns among all vertices of G.

- An $\tilde{O}(|T| + x + k)$ 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.
- An alternative $x = O(k^3)$ 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 $\boldsymbol{x} = \Theta(\boldsymbol{k}^2)$ while the $\boldsymbol{VC}$-dimension argument is limited to showing $\boldsymbol{O}(\boldsymbol{k}^3)$.**

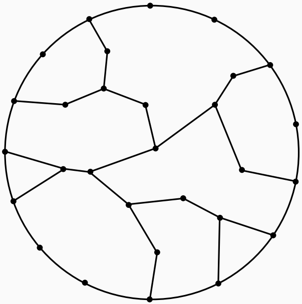
Halin Graphs



Definition

A *Halin graph* is a graph obtained by taking an embedded tree and connecting its leaves by a cycle.

Halin Graphs

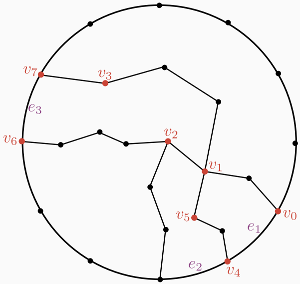


Definition

A *Halin graph* is a graph obtained by taking an embedded tree and connecting its leaves by a cycle.

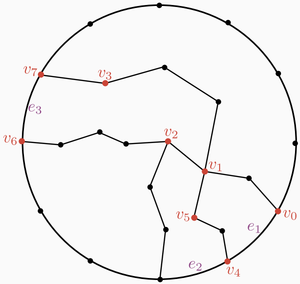
Denote by k the size of the cycle. Consider the patterns of the graph w.r.t. the infinite face.

A Limitation of the VC-dimension Argument



Consider the values of the patterns of $v_0, v_1, \ldots v_7$ at e_1, e_2, e_3:

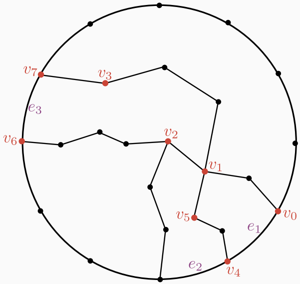
A Limitation of the VC-dimension Argument



Consider the values of the patterns of $v_0, v_1, \ldots v_7$ at e_1, e_2, e_3:

$$\begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{pmatrix} \end{array}$$

A Limitation of the VC-dimension Argument

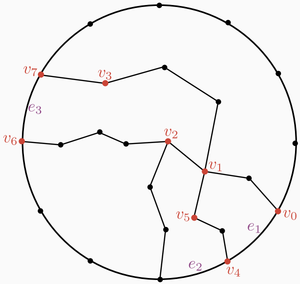


Consider the values of the patterns of $v_0, v_1, \ldots v_7$ at e_1, e_2, e_3:

$$
\begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array}
\begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & -1 \\
1 & -1 & 1 \\
1 & -1 & -1 \\
-1 & 1 & 1 \\
-1 & 1 & -1 \\
-1 & -1 & 1 \\
-1 & -1 & -1
\end{pmatrix} \end{array}
$$

The VC-dimension of the matrix is 3

A Limitation of the VC-dimension Argument



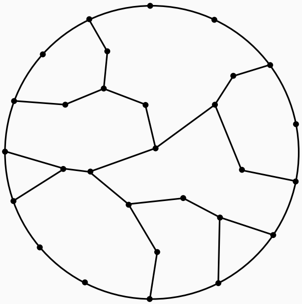
Consider the values of the patterns of $v_0, v_1, \dots v_7$ at e_1, e_2, e_3:

	e_1	e_2	e_3
v_0	1	1	1
v_1	1	1	−1
v_2	1	−1	1
v_3	1	−1	−1
v_4	−1	1	1
v_5	−1	1	−1
v_6	−1	−1	1
v_7	−1	−1	−1

The VC-dimension of the matrix is 3

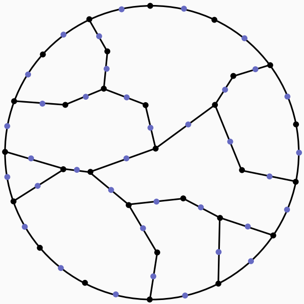
Thus, the VC-dimension argument is limited to showing $O(k^3)$.

An $O(k^2)$ Proof in Halin Graphs



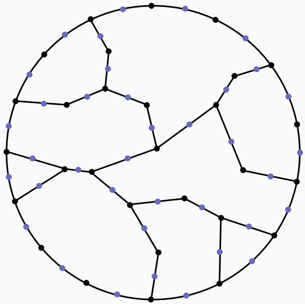
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

An $O(k^2)$ Proof in Halin Graphs



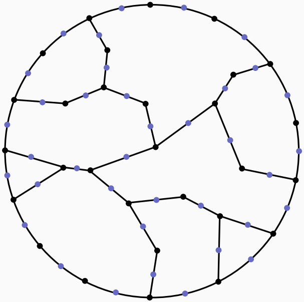
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

An $O(k^2)$ Proof in Halin Graphs



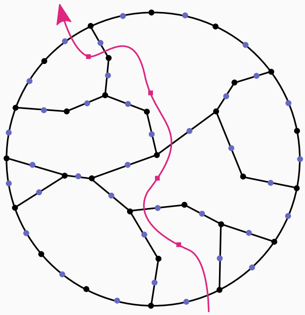
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

An $O(k^2)$ Proof in Halin Graphs



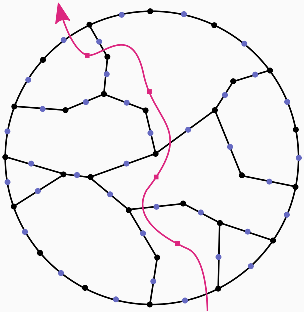
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

An $O(k^2)$ Proof in Halin Graphs



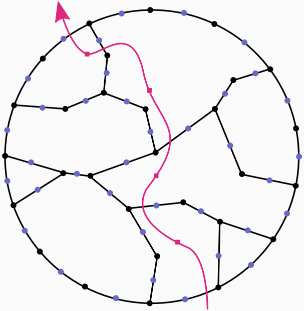
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

An $O(k^2)$ Proof in Halin Graphs



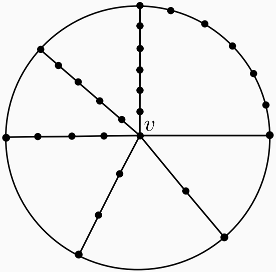
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

An $O(k^2)$ Proof in Halin Graphs



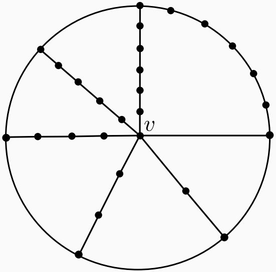
- Subdivide every edge to make patterns binary.
- There are only $O(k)$ vertices of degree > 2, hence $O(k)$ faces in G.
- Every bisector can visit every face at most once.
- The number of distinct patterns along every single face is only $O(k)$.
- Thus, there are only $O(k^2)$ patterns in G.

A matching $\Omega(k^2)$ Lower Bound for Halin Graphs



- Attach $O(k)$ paths of lengths $1, 2, \ldots, \frac{k}{2}$ to a middle vertex v.
- Pad the portions of the infinite face between the first and last paths with $\frac{k}{2}$ vertices.

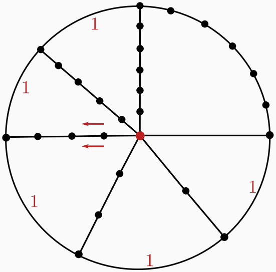
A matching $\Omega(k^2)$ Lower Bound for Halin Graphs



- Attach $O(k)$ paths of lengths $1, 2, \dots, \frac{k}{2}$ to a middle vertex v.
- Pad the portions of the infinite face between the first and last paths with $\frac{k}{2}$ vertices.

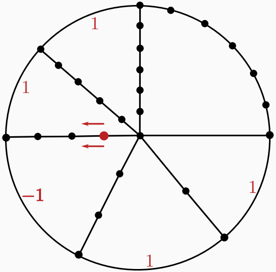
Claim: The path of length i contains $i - 1$ distinct patterns.

The Different Patterns Along a Path



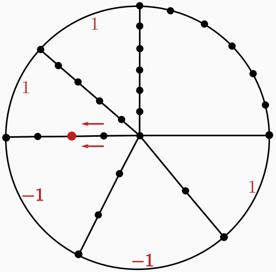
$\langle 1, 1, 1, 1, 1 \rangle$

The Different Patterns Along a Path



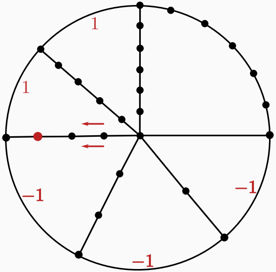
$\langle 1, 1, 1, 1, 1\rangle, \langle 1, 1, -1, 1, 1\rangle$

The Different Patterns Along a Path



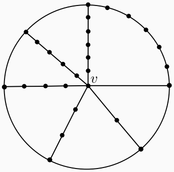
$\langle 1, 1, 1, 1, 1\rangle, \langle 1, 1, -1, 1, 1\rangle, \langle 1, -1, -1, 1, 1\rangle$

The Different Patterns Along a Path



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

A matching $\Omega(k^2)$ Lower Bound for Halin Graphs



The distinct patterns of this graph are thus:

$\langle -1, 1, 1, 1, 1, \ldots \rangle$

$\langle 1, -1, 1, 1, 1, \ldots \rangle, \langle -1, -1, 1, 1, 1, \ldots \rangle$

$\langle 1, 1, -1, 1, 1, \ldots \rangle, \langle 1, -1, -1, 1, 1, \ldots \rangle, \langle -1, -1, -1, 1, 1, \ldots \rangle$

$\langle 1, 1, 1, -1, 1, \ldots \rangle, \langle 1, 1, -1, -1, 1, \ldots \rangle, \langle 1, -1, -1, -1, 1, \ldots \rangle, \langle -1, -1, -1, -1, 1, \ldots \rangle$

$$\vdots$$

$$\sum_{i=1}^{\frac{k}{2}} (i-1) = \Omega(k^2)$$

Open Question 1: Closing the Gap

The number of distinct patterns in planar graphs is $\Omega(k^2)$ and $O(k^3)$.

Open Question 1: Closing the Gap

The number of distinct patterns in planar graphs is $\Omega(k^2)$ and $O(k^3)$.

- To prove an $O(k^2)$ bound: Show that the *total* number of crossings between the bisectors is $O(k^2)$.

Open Question 1: Closing the Gap

The number of distinct patterns in planar graphs is $\Omega(k^2)$ and $O(k^3)$.

- To prove an $O(k^2)$ bound: Show that the *total* number of crossings between the bisectors is $O(k^2)$.
- To prove an $\Omega(k^3)$ bound: Realize an example of k bisectors that cross $\Omega(k^3)$ times.

Open Question 1: Closing the Gap

The number of distinct patterns in planar graphs is $\Omega(k^2)$ and $O(k^3)$.

- To prove an $O(k^2)$ bound: Show that the *total* number of crossings between the bisectors is $O(k^2)$.
- To prove an $\Omega(k^3)$ bound: Realize an example of k bisectors that cross $\Omega(k^3)$ times.

The gap remains difficult even in the family of 2-outerplanar graphs.

Open Question 1: Closing the Gap

The number of distinct patterns in planar graphs is $\Omega(k^2)$ and $O(k^3)$.

- To prove an $O(k^2)$ bound: Show that the *total* number of crossings between the bisectors is $O(k^2)$.
- To prove an $\Omega(k^3)$ bound: Realize an example of k bisectors that cross $\Omega(k^3)$ times.

The gap remains difficult even in the family of 2-outerplanar graphs.

Conjecture

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

Open Question 2: Information-Theoretic Lower Bounds

We showed how to compress the $T \times S$ distances into $\tilde{O}(|T| + x + k)$ bits.

Open Question 2: Information-Theoretic Lower Bounds

We showed how to compress the $T \times S$ distances into $\tilde{O}(|T| + x + k)$ bits.

- A simple argument shows that $\Omega(|T| + k)$ bits are necessary.

Open Question 2: Information-Theoretic Lower Bounds

We showed how to compress the $T \times S$ distances into $\tilde{O}(|T| + x + k)$ bits.

- A simple argument shows that $\Omega(|T| + k)$ bits are necessary.
- Since $x = \Omega(k^2)$, there is an additive gap of at least $\Omega(k^2)$ between our construction and the lower bound.

Open Question 2: Information-Theoretic Lower Bounds

We showed how to compress the $T \times S$ distances into $\tilde{O}(|T| + x + k)$ bits.

- A simple argument shows that $\Omega(|T| + k)$ bits are necessary.
- Since $x = \Omega(k^2)$, there is an additive gap of at least $\Omega(k^2)$ between our construction and the lower bound.
- For small values of $|T|$, the naïve compression of $\tilde{O}(|T| \cdot k)$ bits is still better than ours. Can it be improved?

Open Question 2: Information-Theoretic Lower Bounds

We showed how to compress the $T \times S$ distances into $\tilde{O}(|T| + x + k)$ bits.

- A simple argument shows that $\Omega(|T| + k)$ bits are necessary.
- Since $x = \Omega(k^2)$, there is an additive gap of at least $\Omega(k^2)$ between our construction and the lower bound.
- For small values of $|T|$, the naïve compression of $\tilde{O}(|T| \cdot k)$ bits is still better than ours. Can it be improved?
- Is the additive $\tilde{O}(x)$ bits necessary? It remains open to show a lower bound, or improve the compression to not depend on x.

Open Question 2: Information-Theoretic Lower Bounds

We showed how to compress the $T \times S$ distances into $\tilde{O}(|T| + x + k)$ bits.

- A simple argument shows that $\Omega(|T| + k)$ bits are necessary.
- Since $x = \Omega(k^2)$, there is an additive gap of at least $\Omega(k^2)$ between our construction and the lower bound.
- For small values of $|T|$, the naïve compression of $\tilde{O}(|T| \cdot k)$ bits is still better than ours. Can it be improved?
- Is the additive $\tilde{O}(x)$ bits necessary? It remains open to show a lower bound, or improve the compression to not depend on x.

The End