A logarithmic bound for simultaneous embeddings

Raphael Steiner

ETH zürich

Institute of Theoretical Computer Science
Department of Computer Science
ETH Zürich

Isola delle Femmine, 31st International Symposium on Graph Drawing and Network Visualization
September 21, 2023

Simultaneous embeddability

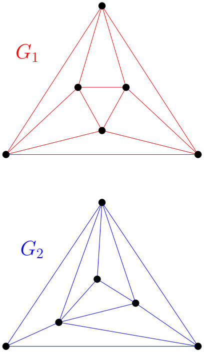
Definition

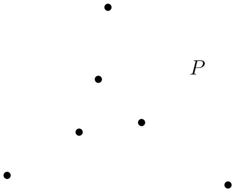
A collection $\mathcal{G}$ of planar graphs on n vertices is *simultaneously embeddable* if there exists a set $P \subseteq \mathbb{R}^2$ of size n such that every $G \in \mathcal{G}$ admits a crossing-free straight-line embedding on P.
Otherwise, $\mathcal{G}$ is a *conflict collection*.

Simultaneous embeddability

Definition

A collection $\mathcal{G}$ of planar graphs on n vertices is *simultaneously embeddable* if there exists a set $P \subseteq \mathbb{R}^2$ of size n such that every $G \in \mathcal{G}$ admits a crossing-free straight-line embedding on P. Otherwise, $\mathcal{G}$ is a *conflict collection*.

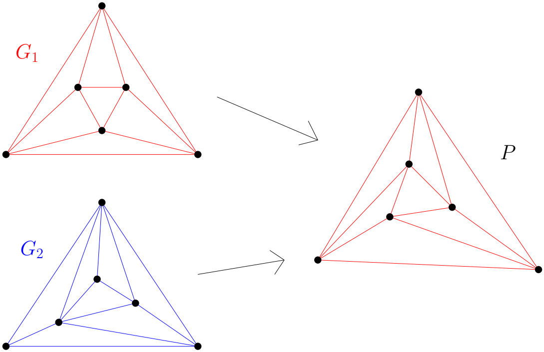




Simultaneous embeddability

Definition

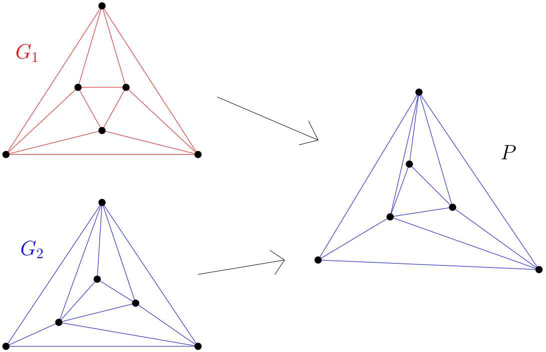
A collection $\mathcal{G}$ of planar graphs on n vertices is *simultaneously embeddable* if there exists a set $P \subseteq \mathbb{R}^2$ of size n such that every $G \in \mathcal{G}$ admits a crossing-free straight-line embedding on P.
Otherwise, $\mathcal{G}$ is a *conflict collection*.



Simultaneous embeddability

Definition

A collection $\mathcal{G}$ of planar graphs on n vertices is *simultaneously embeddable* if there exists a set $P \subseteq \mathbb{R}^2$ of size n such that every $G \in \mathcal{G}$ admits a crossing-free straight-line embedding on P. Otherwise, $\mathcal{G}$ is a *conflict collection*.



Major open problem in the area

Problem (Brass, Cenek, Duncan, Efrat, Erten, Ismailescu, Kobourov, Lubiw and Mitchell 2007)

Is there a conflict collection $\mathcal{G} = \{G_1, G_2\}$ *of size two?*

Progress so far

$\sigma(n) :=$ minimum size of a conflict collection of n-vertex graphs

$\sigma :=$ minimum size of any conflict collection

Progress so far

$\sigma(n)$:=minimum size of a conflict collection of n-vertex graphs

σ :=minimum size of any conflict collection

Problem (Brass, Cenek, Duncan, Efrat, Erten, Ismailescu, Kobourov, Lubiw and Mitchell 2007)

Is $\sigma = 2$?

Progress so far

$\sigma(n) :=$minimum size of a conflict collection of n-vertex graphs
$\sigma :=$minimum size of any conflict collection

Problem (Brass, Cenek, Duncan, Efrat, Erten, Ismailescu, Kobourov, Lubiw and Mitchell 2007)

Is $\sigma = 2$?

- Cardinal, Hoffmann and Kusters (2015): For $n \leq 10$, there is no conflict collection. For $n \geq 15$, we have $\sigma(n) < \infty$.

Progress so far

$\sigma(n)$:=minimum size of a conflict collection of n-vertex graphs

σ :=minimum size of any conflict collection

Problem (Brass, Cenek, Duncan, Efrat, Erten, Ismailescu, Kobourov, Lubiw and Mitchell 2007)

Is $\sigma = 2$?

- Cardinal, Hoffmann and Kusters (2015): For $n \leq 10$, there is no conflict collection. For $n \geq 15$, we have $\sigma(n) < \infty$.
- Cardinal, Hoffmann and Kusters (2015): $\sigma(35) \leq 7393 \Rightarrow \sigma \leq 7393$.

Progress so far

$\sigma(n)$:=minimum size of a conflict collection of n-vertex graphs
σ :=minimum size of any conflict collection

Problem (Brass, Cenek, Duncan, Efrat, Erten, Ismailescu, Kobourov, Lubiw and Mitchell 2007)

Is $\sigma = 2$?

- Cardinal, Hoffmann and Kusters (2015): For $n \leq 10$, there is no conflict collection. For $n \geq 15$, we have $\sigma(n) < \infty$.
- Cardinal, Hoffmann and Kusters (2015): $\sigma(35) \leq 7393 \Rightarrow \sigma \leq 7393$.
- Scheucher, Schrezenmaier and S. (2019): $\sigma(11) \leq 49 \Rightarrow \sigma \leq 49$.

Progress so far

$\sigma(n)$:=minimum size of a conflict collection of n-vertex graphs
σ :=minimum size of any conflict collection

Problem (Brass, Cenek, Duncan, Efrat, Erten, Ismailescu, Kobourov, Lubiw and Mitchell 2007)

Is $\sigma = 2$?

- Cardinal, Hoffmann and Kusters (2015): For $n \leq 10$, there is no conflict collection. For $n \geq 15$, we have $\sigma(n) < \infty$.
- Cardinal, Hoffmann and Kusters (2015): $\sigma(35) \leq 7393 \Rightarrow \sigma \leq 7393$.
- Scheucher, Schrezenmaier and S. (2019): $\sigma(11) \leq 49 \Rightarrow \sigma \leq 49$.
- Goenka, Semnani and Yip (2023): $\sigma(n) = O(1.135^n)$ via an explicit construction of a conflict collection.

New results

Theorem (S. 2023)

It holds that $\sigma(n) \leq (3 + o(1)) \log_2(n)$.

New results

Theorem (S. 2023)

It holds that $\sigma(n) \leq (3 + o(1)) \log_2(n)$.

Theorem (S. 2023)

For every $n \in \{107, \ldots, 193\}$, *we have* $\sigma(n) \leq 30 \Rightarrow \sigma \leq 30$.

New results

Theorem (S. 2023)

It holds that $\sigma(n) \leq (3 + o(1)) \log_2(n)$.

Theorem (S. 2023)

For every $n \in \{107, \ldots, 193\}$, *we have* $\sigma(n) \leq 30 \Rightarrow \sigma \leq 30$.

Theorem (S. 2023)

For $n \geq 5040$ *there exists an explicitly constructed conflict collection consisting of*

$$n^{\underline{6}} + 1 = n(n-1)(n-2)(n-3)(n-4)(n-5) + 1 < n^6$$

planar n-vertex graphs.

Randomly generating stacked triangulations

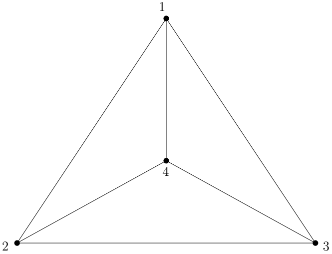
Random process to generate n-vertex labelled stacked triangulation:

- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i-4$ faces of the current stacked triangulation and stack a vertex with label $i+1$ into the selected face.

Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

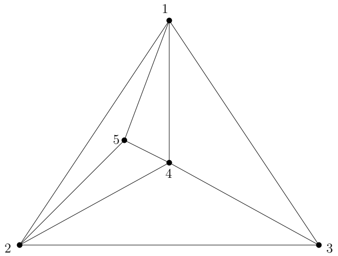
- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i-4$ faces of the current stacked triangulation and stack a vertex with label $i+1$ into the selected face.



Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

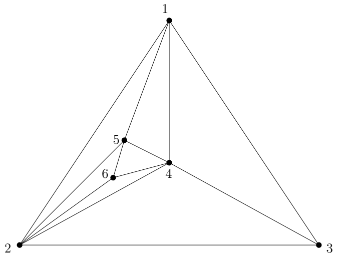
- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i - 4$ faces of the current stacked triangulation and stack a vertex with label $i + 1$ into the selected face.



Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

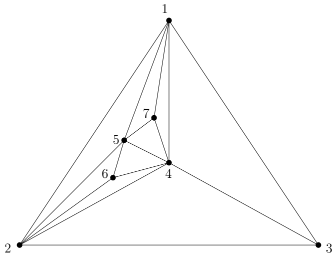
- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i - 4$ faces of the current stacked triangulation and stack a vertex with label $i + 1$ into the selected face.



Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

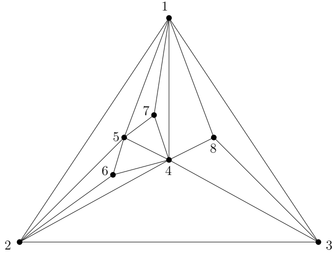
- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i - 4$ faces of the current stacked triangulation and stack a vertex with label $i + 1$ into the selected face.



Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

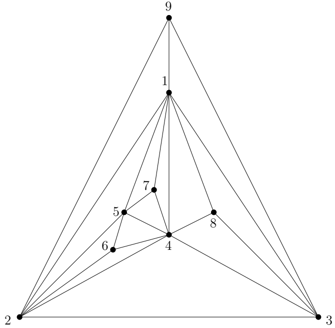
- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i-4$ faces of the current stacked triangulation and stack a vertex with label $i+1$ into the selected face.



Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i - 4$ faces of the current stacked triangulation and stack a vertex with label $i + 1$ into the selected face.



Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \ldots, n-1$, randomly and uniformly select one of the $2i - 4$ faces of the current stacked triangulation and stack a vertex with label $i + 1$ into the selected face.

Important to note:

Randomly generating stacked triangulations

Random process to generate n-vertex labelled stacked triangulation:

- Start from a K_4 on vertices $\{1, 2, 3, 4\}$.
- For $i = 4, \dots, n-1$, randomly and uniformly select one of the $2i-4$ faces of the current stacked triangulation and stack a vertex with label $i+1$ into the selected face.

Important to note:

- Number of possible outcomes of the process:

$$4 \cdot 6 \cdots (2(n-1)-4) = 2^{n-4}(n-3)!$$

- Each of them is equally likely
- Given a labelling of a point set P, at most one stacked triangulation embeds in a label-preserving way

Main lemma

Lemma

Let **G** *denote the random n-vertex triangulation generated according to the described process. Then for every* $P \subseteq \mathbb{R}^2$ *of size n, we have:*

$$\mathbb{P}(\mathbf{G} \text{ embeds straight-line on } P) \leq \frac{16n(n-1)(n-2)}{2^n} = 2^{-(1-o(1))n}$$

Main lemma

Lemma

Let $\mathbf{G}$ *denote the random n-vertex triangulation generated according to the described process. Then for every* $P \subseteq \mathbb{R}^2$ *of size n, we have:*

$$\mathbb{P}(\mathbf{G} \textit{ embeds straight-line on } P) \leq \frac{16n(n-1)(n-2)}{2^n} = 2^{-(1-o(1))n}$$

Proof.

Every straight-line embedding of $\mathbf{G}$ induces a $\{1, \ldots, n\}$-labelling of P. There are $n!$ such labellings. For a fixed labelling of P, at most one triangulation embeds in label-preserving way. Thus, at most $n!$ of the relevant stacked triangulations embed on P. Hence,

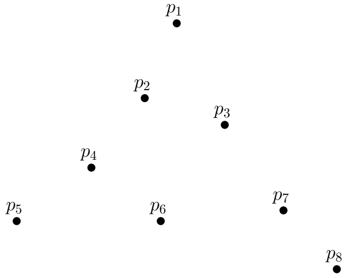
$$\mathbb{P}(\mathbf{G} \text{ embeds on } P) \leq \frac{n!}{2^{n-4}(n-3)!} = \frac{16n(n-1)(n-2)}{2^n}.$$

□

Straight-line embeddability and order types

Definition

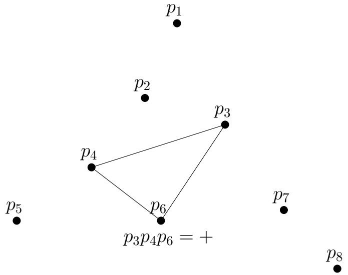
Two point sets $P = \{p_1, \ldots, p_n\}, Q = \{q_1, \ldots, q_n\}$have same order-type if $\forall i, j, k$: $p_i p_j p_k$ and $q_i q_j q_k$ have the same orientation.



Straight-line embeddability and order types

Definition

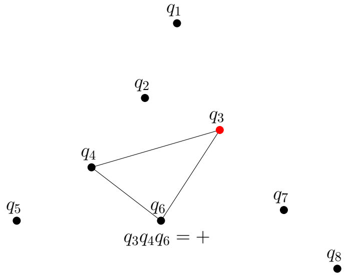
Two point sets $P = \{p_1, \ldots, p_n\}, Q = \{q_1, \ldots, q_n\}$have same order-type if $\forall i, j, k$: $p_i p_j p_k$ and $q_i q_j q_k$ have the same orientation.



Straight-line embeddability and order types

Definition

Two point sets $P = \{p_1, \ldots, p_n\}, Q = \{q_1, \ldots, q_n\}$have same order-type if $\forall i, j, k$: $p_i p_j p_k$ and $q_i q_j q_k$ have the same orientation.



Straight-line embeddability and order types

Observation

If P and Q have the same order type, then a planar graph G embeds on P iff it embeds on Q.

Straight-line embeddability and order types

Observation

If P and Q have the same order type, then a planar graph G embeds on P iff it embeds on Q.

Theorem (Alon 1986)

There are $n^{(4+o(1))n}$ labelled order types of n points in the plane.

Straight-line embeddability and order types

Observation

If P and Q have the same order type, then a planar graph G embeds on P iff it embeds on Q.

Theorem (Alon 1986)

There are $n^{(4+o(1))n}$ labelled order types of n points in the plane.

Corollary

There exists a collection $\mathcal{P}_n$ consisting of n-point sets with $|\mathcal{P}_n| = n^{(4+o(1))n}$ such that the following holds. If planar graphs $G_1, \ldots, G_k$ are simultaneously embeddable, then there is $P \in \mathcal{P}_n$ such that every G_i embeds on P.

Proof of $\sigma(n) \leq (4+o(1))\log_2(n)$.

Theorem

For all $\varepsilon > 0$ *there is* $n_0 = n_0(\varepsilon) \in \mathbb{N}$ *such that* $\sigma(n) \leq (4+\varepsilon)\log_2(n)$ *for all* $n \geq n_0$.

Proof of $\sigma(n) \leq (4 + o(1)) \log_2(n)$.

Theorem

For all $\varepsilon > 0$ there is $n_0 = n_0(\varepsilon) \in \mathbb{N}$ such that $\sigma(n) \leq (4 + \varepsilon) \log_2(n)$ for all $n \geq n_0$.

Proof Sketch.

Let $k = \lfloor (4 + \varepsilon) \log_2(n) \rfloor$. Consider k independently generated random triangulations $\{\mathbf{G}_1, \ldots, \mathbf{G}_k\}$. Then for every $P \in \mathcal{P}_n$:

$$\mathbb{P}\left(\bigwedge_{i=1}^{k} \{\mathbf{G}_i \text{ embeds on } P\}\right) \leq \left(2^{-(1-o(1))n}\right)^k = 2^{-(1-o(1))kn}.$$

Proof of $\sigma(n) \leq (4+o(1))\log_2(n)$.

Theorem

For all $\varepsilon > 0$ there is $n_0 = n_0(\varepsilon) \in \mathbb{N}$ such that $\sigma(n) \leq (4+\varepsilon)\log_2(n)$ for all $n \geq n_0$.

Proof Sketch.

Let $k = \lfloor (4+\varepsilon)\log_2(n) \rfloor$. Consider k independently generated random triangulations $\{\mathbf{G}_1, \ldots, \mathbf{G}_k\}$. Then for every $P \in \mathcal{P}_n$:

$$\mathbb{P}\left(\bigwedge_{i=1}^{k} \{\mathbf{G}_i \text{ embeds on } P\}\right) \leq (2^{-(1-o(1))n})^k = 2^{-(1-o(1))kn}.$$

$$\mathbb{P}(\{\mathbf{G}_1, \ldots, \mathbf{G}_k\} \text{ is simult. embedd.}) \leq |\mathcal{P}_n| \cdot 2^{-(1-o(1))kn}$$

Proof of $\sigma(n) \leq (4 + o(1)) \log_2(n)$.

Theorem

For all $\varepsilon > 0$ *there is* $n_0 = n_0(\varepsilon) \in \mathbb{N}$ *such that* $\sigma(n) \leq (4 + \varepsilon) \log_2(n)$ *for all* $n \geq n_0$.

Proof Sketch.

Let $k = \lfloor (4 + \varepsilon) \log_2(n) \rfloor$. Consider k independently generated random triangulations $\{\mathbf{G}_1, \dots, \mathbf{G}_k\}$. Then for every $P \in \mathcal{P}_n$:

$$\mathbb{P}\left(\bigwedge_{i=1}^{k} \{\mathbf{G}_i \text{ embeds on } P\}\right) \leq (2^{-(1-o(1))n})^k = 2^{-(1-o(1))kn}.$$

$$\mathbb{P}(\{\mathbf{G}_1, \dots, \mathbf{G}_k\} \text{ is simult. embedd.}) \leq |\mathcal{P}_n| \cdot 2^{-(1-o(1))kn}$$

$$= n^{(4+o(1))n} \cdot 2^{-(4+\varepsilon-o(1))n \log_2(n)} = 2^{-(\varepsilon-o(1))n \log_2(n)} \to 0.$$

□

The end

Thank you for your attention!