Counting and Enumerating Crossing-free Perfect Matchings

Manuel Wettstein

ETH Zürich

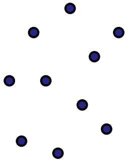
February 21, 2014

Introduction

Crossing-free Perfect Matching: A set of $n/2$ non-intersecting segments with endpoints in a set P of n points.

Introduction

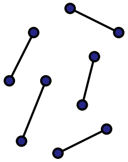
Crossing-free Perfect Matching: A set of $n/2$ non-intersecting segments with endpoints in a set P of n points.



P

Introduction

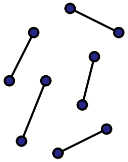
Crossing-free Perfect Matching: A set of $n/2$ non-intersecting segments with endpoints in a set P of n points.



$G \in \mathcal{M}(P)$

Introduction

Crossing-free Perfect Matching: A set of $n/2$ non-intersecting segments with endpoints in a set P of n points.

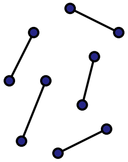


$G \in \mathcal{M}(P)$

- **Lower bound:** $|\mathcal{M}(P)| = \Omega^*(2^n)$ for all P. [García, Noy, Tejel 00]

Introduction

Crossing-free Perfect Matching: A set of $n/2$ non-intersecting segments with endpoints in a set P of n points.

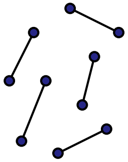


$G \in \mathcal{M}(P)$

- **Lower bound:** $|\mathcal{M}(P)| = \Omega^*(2^n)$ for all P. [García, Noy, Tejel 00]
- **Enumeration:** In polynomial time per enumerated graph. [Wettstein 14]

Introduction

Crossing-free Perfect Matching: A set of $n/2$ non-intersecting segments with endpoints in a set P of n points.

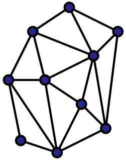


$G \in \mathcal{M}(P)$

- **Lower bound:** $|\mathcal{M}(P)| = \Omega^*(2^n)$ for all P. [García, Noy, Tejel 00]
- **Enumeration:** In polynomial time per enumerated graph. [Wettstein 14]
- **Counting:** In time $O^*(2^n)$. [Wettstein 14]

Introduction

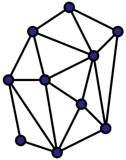
Triangulation: An edge-maximal crossing-free geometric graph.



$G \in \mathcal{T}(P)$

Introduction

Triangulation: An edge-maximal crossing-free geometric graph.

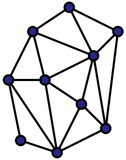


$G \in \mathcal{T}(P)$

- **Lower bound:** $|\mathcal{T}(P)| = \Omega^*(2.43^n)$ for all P. [Sharir, Sheffer, Welzl 10]

Introduction

Triangulation: An edge-maximal crossing-free geometric graph.

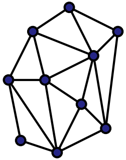


$G \in \mathcal{T}(P)$

- **Lower bound:** $|\mathcal{T}(P)| = \Omega^*(2.43^n)$ for all P. [Sharir, Sheffer, Welzl 10]
- **Enumeration:** In polynomial time per enumerated graph. [Avis, Fukuda 96]

Introduction

Triangulation: An edge-maximal crossing-free geometric graph.



$G \in \mathcal{T}(P)$

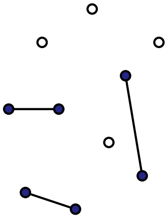
- **Lower bound:** $|\mathcal{T}(P)| = \Omega^*(2.43^n)$ for all P. [Sharir, Sheffer, Welzl 10]
- **Enumeration:** In polynomial time per enumerated graph. [Avis, Fukuda 96]
- **Counting:** In time $O^*(2^n)$. [Seidel, Alvarez 13]

Counting Crossing-free Perfect Matchings

Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.

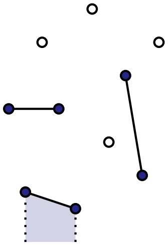
Counting Crossing-free Perfect Matchings

Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.



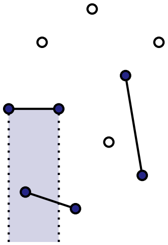
Counting Crossing-free Perfect Matchings

Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.



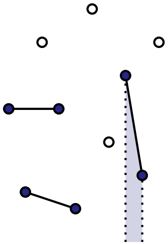
Counting Crossing-free Perfect Matchings

Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.



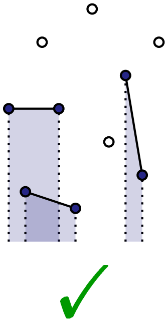
Counting Crossing-free Perfect Matchings

Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.



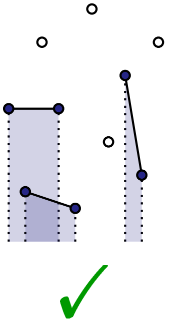
Counting Crossing-free Perfect Matchings

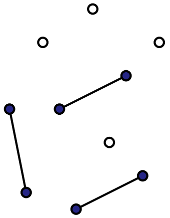
Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.



Counting Crossing-free Perfect Matchings

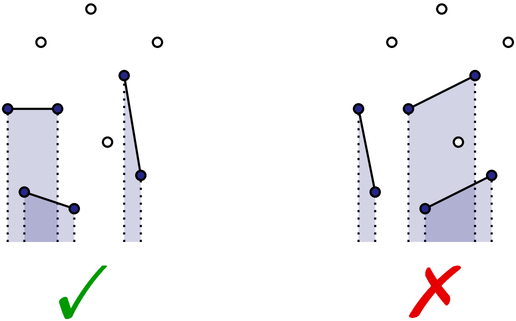
Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.





Counting Crossing-free Perfect Matchings

Lower crossing-free matching (LCM): A crossing-free matching where no unmatched point is contained in the lower shadow of a segment.

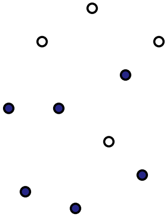


Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.

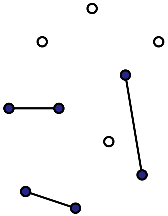
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.



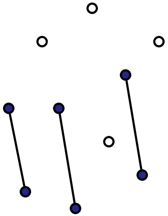
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.



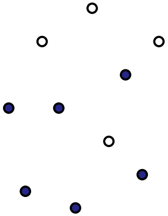
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.



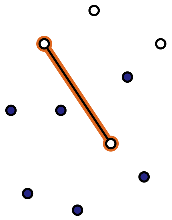
Counting Crossing-free Perfect Matchings

- **Configuration:** A {●, ○}-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints ○ that contains no ○ in its lower shadow and no ● in its upper shadow.



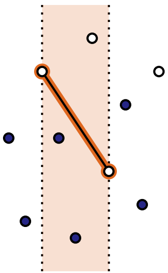
Counting Crossing-free Perfect Matchings

- **Configuration:** A {●, ○}-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints ○ that contains no ○ in its lower shadow and no ● in its upper shadow.



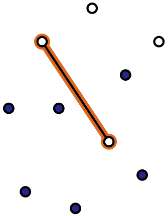
Counting Crossing-free Perfect Matchings

- **Configuration:** A {●, ○}-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints ○ that contains no ○ in its lower shadow and no ● in its upper shadow.



Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints $\circ$ that contains no $\circ$ in its lower shadow and no $\bullet$ in its upper shadow.

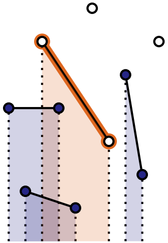


Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints $\circ$ that contains no $\circ$ in its lower shadow and no $\bullet$ in its upper shadow.

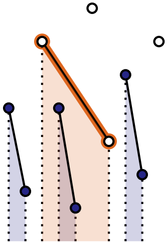


Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints $\circ$ that contains no $\circ$ in its lower shadow and no $\bullet$ in its upper shadow.

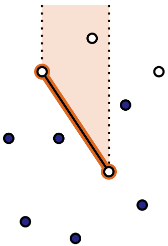


Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints ○ that contains no ○ in its lower shadow and no ● in its upper shadow.

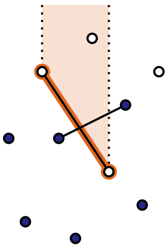


Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A {●, ○}-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints ○ that contains no ○ in its lower shadow and no ● in its upper shadow.

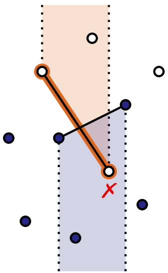


Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints $\circ$ that contains no $\circ$ in its lower shadow and no $\bullet$ in its upper shadow.

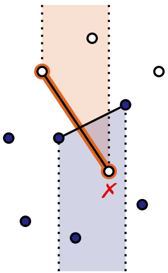


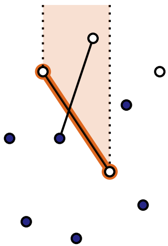
Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints ∘ that contains no ∘ in its lower shadow and no • in its upper shadow.



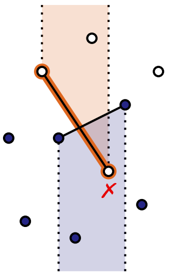


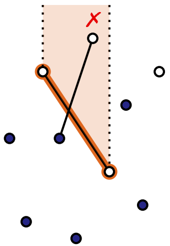
Claim

Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P, representing a set of *compatible* LCMs.
- **Advance (for LCM):** A segment with endpoints $\circ$ that contains no $\circ$ in its lower shadow and no $\bullet$ in its upper shadow.





Claim

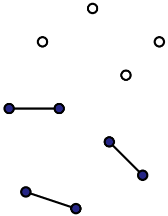
Adding an advance to a compatible LCM results in a larger LCM.

Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.

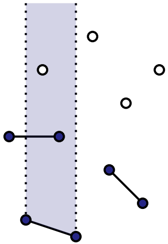
Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.



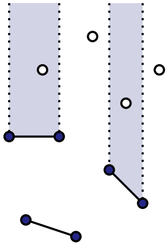
Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.



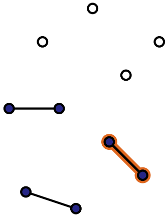
Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.



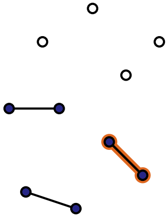
Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.



Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.

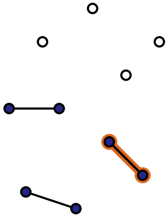


Claims

Every non-empty LCM has a unique extreme segment.

Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.

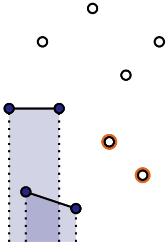


Claims

Every non-empty LCM has a unique extreme segment.
Removing it results in a smaller LCM.

Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.

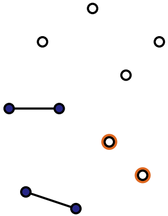


Claims

Every non-empty LCM has a unique extreme segment.
Removing it results in a smaller LCM.

Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.

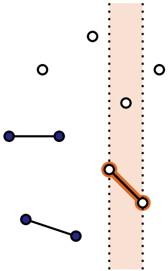


Claims

Every non-empty LCM has a unique extreme segment.
Removing it results in a smaller LCM.
The removed segment is an advance for the obtained smaller LCM.

Counting Crossing-free Perfect Matchings

- **Extreme segment (in LCM):** The right-most segment that contains no (parts of another) segment in its upper shadow.



Claims

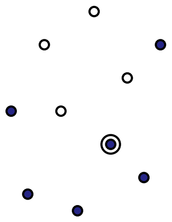
Every non-empty LCM has a unique extreme segment.
Removing it results in a smaller LCM.
The removed segment is an advance for the obtained smaller LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.

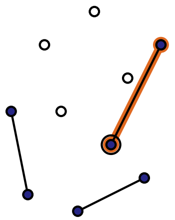
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.



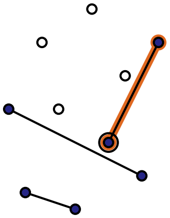
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.



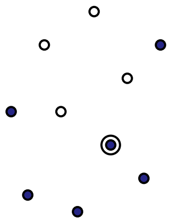
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.



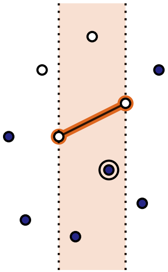
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.



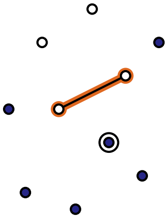
Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.



Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.

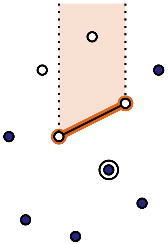


Claim

An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.

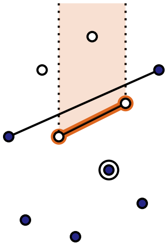


Claim

An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.

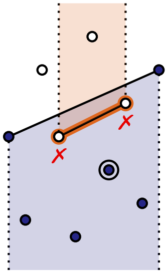


Claim

An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.

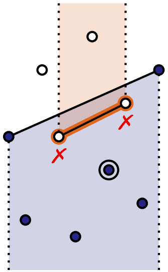


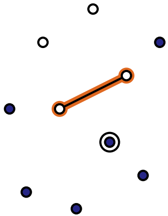
Claim

An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A $\{\bullet, \circ\}$-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.



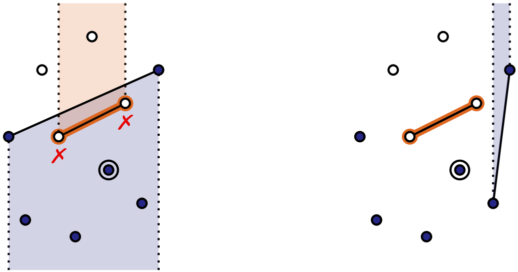


Claim

An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A {●, ○}-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.

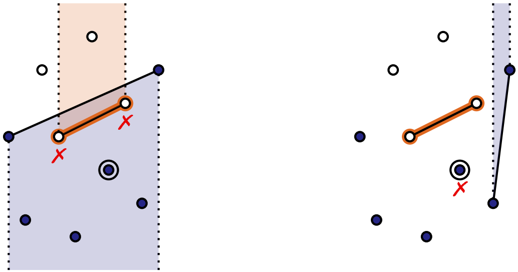


Claim

An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Configuration:** A {●, ○}-coloring of P with a marking on exactly one matched point.
- **Advance (for LCM):** A segment as before, with the additional constraint that its right endpoint is to the right of the marking.



Claim

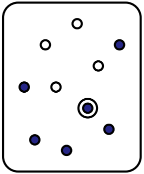
An advance, when added to a compatible LCM, is always the extreme segment of the resulting larger LCM.

Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.

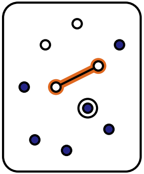
Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.



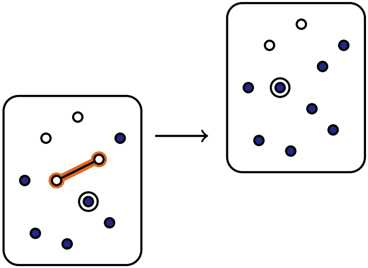
Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.



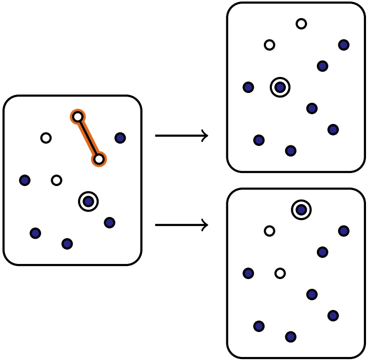
Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.



Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.

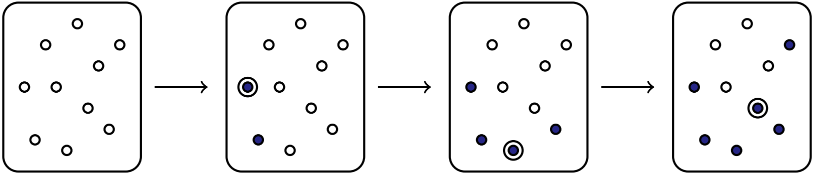


Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.
- **Rooted Path (in Γ_P):** Directed path which starts in the configuration without any matched points.

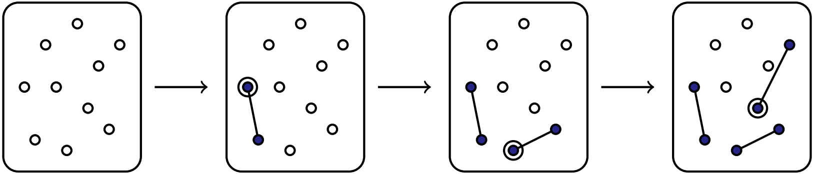
Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.
- **Rooted Path (in Γ_P):** Directed path which starts in the configuration without any matched points.



Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.
- **Rooted Path (in Γ_P):** Directed path which starts in the configuration without any matched points.



Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.
- **Rooted Path (in Γ_P):** Directed path which starts in the configuration without any matched points.

Claim

There is a natural bijection between rooted paths in Γ_P and LCMs on P.

Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.
- **Rooted Path (in Γ_P):** Directed path which starts in the configuration without any matched points.

Claim

There is a natural bijection between rooted paths in Γ_P and LCMs on P.

Claim

The advance graph Γ_P has at most $2^n n$ vertices and at most $2^n n^3$ edges.

Counting Crossing-free Perfect Matchings

- **Advance Graph (Γ_P):** A directed acyclic graph over configurations, where edges correspond to advances.
- **Rooted Path (in Γ_P):** Directed path which starts in the configuration without any matched points.

Claim

There is a natural bijection between rooted paths in Γ_P and LCMs on P.

Claim

The advance graph Γ_P has at most $2^n n$ vertices and at most $2^n n^3$ edges.

Theorem

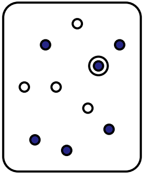
The number $|\mathcal{M}(P)|$ can be computed in time $O(n^3 2^n)$.

Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.

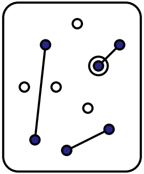
Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.



Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.



Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.

Solution 1:

- Compute an explicit representation of Γ_P in time $O^*(2^n)$.

Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.

Solution 1:

- Compute an explicit representation of Γ_P in time $O^*(2^n)$.
- Remove all dead ends in Γ_P in time $O^*(2^n)$.

Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.

Solution 1:

- Compute an explicit representation of Γ_P in time $O^*(2^n)$.
- Remove all dead ends in Γ_P in time $O^*(2^n)$.
- Use this nice representation of Γ_P to enumerate $\mathcal{M}(P)$.

Enumerating Crossing-free Perfect Matchings

Problem: There are many dead ends in Γ_P.

Solution 1:

- Compute an explicit representation of Γ_P in time $O^*(2^n)$.
- Remove all dead ends in Γ_P in time $O^*(2^n)$.
- Use this nice representation of Γ_P to enumerate $\mathcal{M}(P)$.

Theorem

The set $\mathcal{M}(P)$ can be enumerated efficiently, that is, in time $O^*(2^n + |\mathcal{M}(P)|) = O^*(|\mathcal{M}(P)|)$.

Enumerating Crossing-free Perfect Matchings

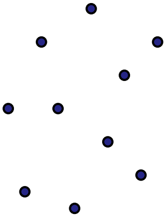
Solution 2:

- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.

Enumerating Crossing-free Perfect Matchings

Solution 2:

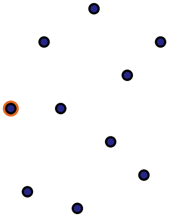
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

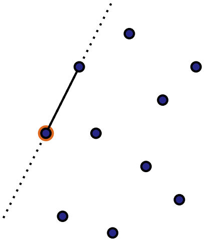
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

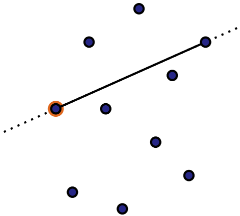
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

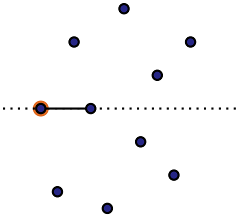
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

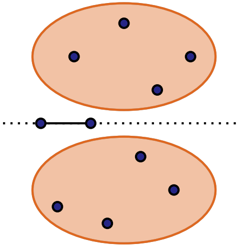
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

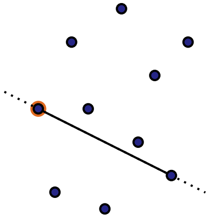
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

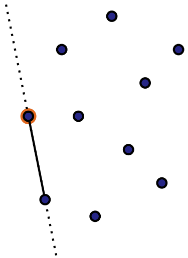
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

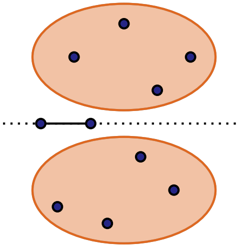
- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



Enumerating Crossing-free Perfect Matchings

Solution 2:

- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



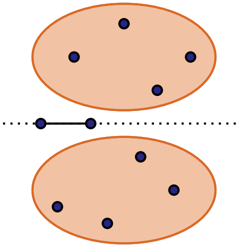
Let $e_m :=$ #easy perfect matchings on $2m = n$ points.

$$e_0 = 1, e_m = \sum_{i=0}^{m-1} e_i \cdot e_{m-i-1} \quad \Rightarrow \quad e_m = \Theta^*(4^m) = \Theta^*(2^n)$$

Enumerating Crossing-free Perfect Matchings

Solution 2:

- During the preprocessing stage, enumerate $\Theta^*(2^n)$ "easy" crossing-free perfect matchings.



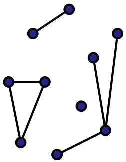
Let $e_m :=$ #easy perfect matchings on $2m = n$ points.

$$e_0 = 1, e_m = \sum_{i=0}^{m-1} e_i \cdot e_{m-i-1} \quad \Rightarrow \quad e_m = \Theta^*(4^m) = \Theta^*(2^n)$$

Theorem

The set $\mathcal{M}(P)$ can be enumerated with polynomial time delay per enumerated object.

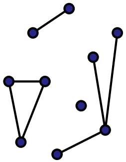
All Crossing-free Geometric Graphs



$G \in \mathcal{G}(P)$

- **Lower bound:** $|\mathcal{G}(P)| = \Omega^*(11.65^n)$ for all P. [Flajolet, Noy 99]

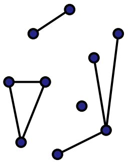
All Crossing-free Geometric Graphs



$G \in \mathcal{G}(P)$

- **Lower bound:** $|\mathcal{G}(P)| = \Omega^*(11.65^n)$ for all P. [Flajolet, Noy 99]
- **Enumeration:** In polynomial time per enumerated graph. [Aichholzer et al. 07]

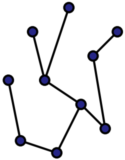
All Crossing-free Geometric Graphs



$G \in \mathcal{G}(P)$

- **Lower bound:** $|\mathcal{G}(P)| = \Omega^*(11.65^n)$ for all P. [Flajolet, Noy 99]
- **Enumeration:** In polynomial time per enumerated graph. [Aichholzer et al. 07]
- **Counting:** In time $O^*(2.839^n)$. [Wettstein 14]

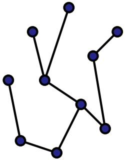
Crossing-free Spanning Trees



$G \in \mathcal{ST}(P)$

- **Lower bound:** $|\mathcal{ST}(P)| = \Omega^*(6.75^n)$ for all P. [Flajolet, Noy 99]

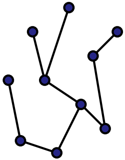
Crossing-free Spanning Trees



$G \in \mathcal{ST}(P)$

- **Lower bound:** $|\mathcal{ST}(P)| = \Omega^*(6.75^n)$ for all P. [Flajolet, Noy 99]
- **Enumeration:** In polynomial time per enumerated graph. [Avis, Fukuda 96]

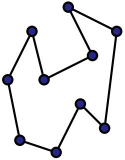
Crossing-free Spanning Trees



$G \in \mathcal{ST}(P)$

- **Lower bound:** $|\mathcal{ST}(P)| = \Omega^*(6.75^n)$ for all P. [Flajolet, Noy 99]
- **Enumeration:** In polynomial time per enumerated graph. [Avis, Fukuda 96]
- **Counting:** In time $O^*(c^n)$ for $c < 8$. [Wettstein 14]

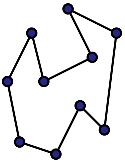
Crossing-free Spanning Cycles



$G \in \mathcal{SC}(P)$

- **Lower bound:** $|\mathcal{SC}(P)| \geq 1$ for all P.

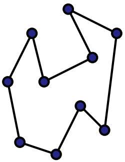
Crossing-free Spanning Cycles



$G \in \mathcal{SC}(P)$

- **Lower bound:** $|\mathcal{SC}(P)| \geq 1$ for all P.
- **Enumeration:** –

Crossing-free Spanning Cycles



$G \in \mathcal{SC}(P)$

- **Lower bound:** $|\mathcal{SC}(P)| \geq 1$ for all P.
- **Enumeration:** –
- **Counting:** In time $O^*(c^n)$ for $c < 6$. [Wettstein 14]

Thanks for your attention!