Mittagsseminar

Algorithms for sampling random pseudoline arrangements

Sandro M. Roch

16.12.2021

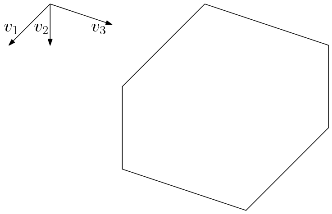
Rhombic tilings of zonotopes

- **Def:** $v_1, \ldots, v_r \in \mathbb{R}^2$ pw. indep. define *zonotope*

$$Z(v_1, \ldots, v_r) := \left\{ \sum \lambda_i v_i : \lambda_i \in [-1, 1] \right\} .$$

- **Def:** $v_1, \ldots, v_r \in \mathbb{R}^2$ pw. indep. define *zonotope*

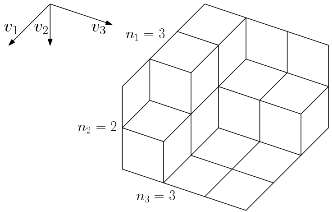
$$Z(v_1, \ldots, v_r) := \left\{ \sum \lambda_i v_i : \lambda_i \in [-1, 1] \right\} .$$



Rhombic tilings of zonotopes

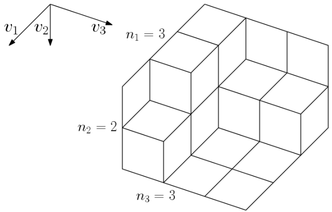
- **Def:** $v_1, \ldots, v_r \in \mathbb{R}^2$ pw. indep. define *zonotope*

$$Z(v_1, \ldots, v_r) := \left\{ \sum \lambda_i v_i : \lambda_i \in [-1, 1] \right\} .$$



- **Def:** $v_1, \ldots, v_r \in \mathbb{R}^2$ pw. indep. define *zonotope*

$$Z(v_1, \ldots, v_r) := \left\{ \sum \lambda_i v_i : \lambda_i \in [-1, 1] \right\} .$$

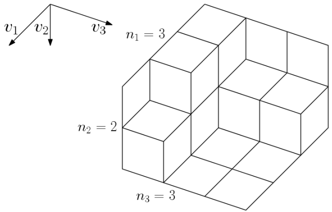


- Zonotopes $Z(v_i, v_j)$, $v_i \neq v_j$ are *rhombi* (also *lozenges*).

Rhombic tilings of zonotopes

- **Def:** $v_1, \ldots, v_r \in \mathbb{R}^2$ pw. indep. define *zonotope*

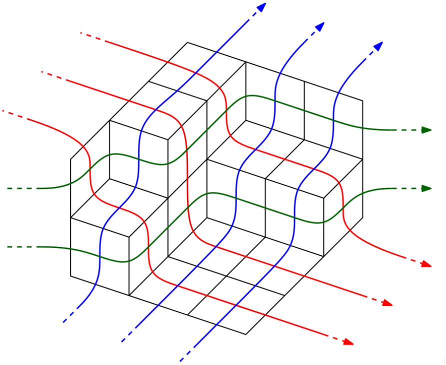
$$Z(v_1, \ldots, v_r) := \left\{ \sum \lambda_i v_i : \lambda_i \in [-1, 1] \right\} .$$



- Zonotopes $Z(v_i, v_j)$, $v_i \neq v_j$ are *rhombi* (also *lozenges*).
- *Rhombic tiling*: For fixed *shape* $n_1, \ldots, n_r \in \mathbb{N}$, tiling of $Z(v_1, \ldots, v_r)$ by rhombi of type $Z\left(\frac{v_i}{n_i}, \frac{v_j}{n_j}\right)$.

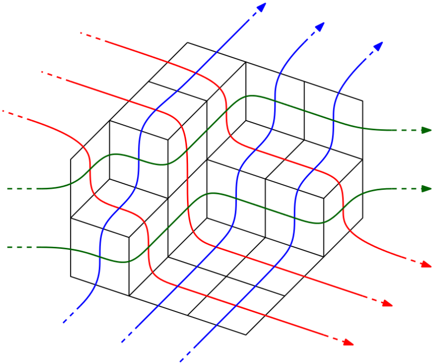
Rhombic tilings and pseudoline arrangements

- Draw curves through ribbons of parallel edges:



Rhombic tilings and pseudoline arrangements

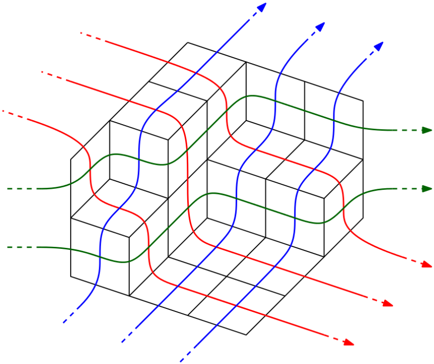
- Draw curves through ribbons of parallel edges:



- Yields „generalized“ *simple pseudoline arrangement*:
  - r classes of $n_1, \ldots, n_r$ non-intersecting pseudolines.
  - Pseudolines of different classes cross each other exactly once.

Rhombic tilings and pseudoline arrangements

- Draw curves through ribbons of parallel edges:



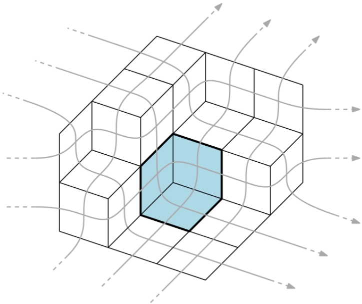
- Yields „generalized“*simple pseudoline arrangement*:
  - r classes of $n_1, \ldots, n_r$ non-intersecting pseudolines.
  - Pseudolines of different classes cross each other exactly once.
- Generalizable to non-simple arrangements

Uniform sampling of pseudoline arrangements

- **Big open problem:** Polynomial algorithm for uniform sampling PLA.
  - Motivation: *Quasicrytals* in physics
  - Motivation: Determine average characteristics
  - Motivation: Approximate counting
  - Motivation: Islamic art (Adding symmetry)

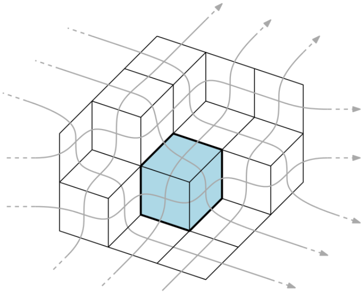
Uniform sampling of pseudoline arrangements

- **Big open problem:** Polynomial algorithm for uniform sampling PLA.
  - Motivation: *Quasicrytals* in physics
  - Motivation: Determine average characteristics
  - Motivation: Approximate counting
  - Motivation: Islamic art (Adding symmetry)
- **Idea:** Use Markov chains with *flips* as transitions:



Uniform sampling of pseudoline arrangements

- **Big open problem:** Polynomial algorithm for uniform sampling PLA.
  - Motivation: *Quasicrytals* in physics
  - Motivation: Determine average characteristics
  - Motivation: Approximate counting
  - Motivation: Islamic art (Adding symmetry)
- **Idea:** Use Markov chains with *flips* as transitions:



Rapidly mixing Markov chains

Markov chain (X_t), state space $\mathcal{X}$, transition prob. $P : \mathcal{X} \times \mathcal{X} \to [0, 1]$

- If *irreducible* and *aperiodic*: Converges to *stationary distribution* π:

$$\mathbb{P}[X_t = x] \to \pi(x)$$

Rapidly mixing Markov chains

Markov chain (X_t), state space $\mathcal{X}$, transition prob. $P : \mathcal{X} \times \mathcal{X} \to [0, 1]$

- If *irreducible* and *aperiodic*: Converges to *stationary distribution* π:

$$\mathbb{P}[X_t = x] \to \pi(x)$$

- If P symmetric: π equals uniform distribution.

Rapidly mixing Markov chains

Markov chain (X_t), state space $\mathcal{X}$, transition prob. $P : \mathcal{X} \times \mathcal{X} \to [0, 1]$

- If *irreducible* and *aperiodic*: Converges to *stationary distribution* π:

$$\mathbb{P}[X_t = x] \to \pi(x)$$

- If P symmetric: π equals uniform distribution.
- *Total variation distance* of distributions μ, μ' on $\mathcal{X}$:

$$\|\mu - \mu'\|_{\mathsf{TV}} := \sup_{M \subseteq \mathcal{X}} |\mu(M) - \mu'(M)|$$

Rapidly mixing Markov chains

Markov chain (X_t), state space $\mathcal{X}$, transition prob. $P : \mathcal{X} \times \mathcal{X} \to [0,1]$

- If *irreducible* and *aperiodic*: Converges to *stationary distribution* π:

$$\mathbb{P}[X_t = x] \to \pi(x)$$

- If P symmetric: π equals uniform distribution.
- *Total variation distance* of distributions μ, μ' on $\mathcal{X}$:

$$\|\mu - \mu'\|_{\mathsf{TV}} := \sup_{M \subseteq \mathcal{X}} |\mu(M) - \mu'(M)|$$

- Define $d(t) := \max_{x \in \mathcal{X}} \|P^t(x, \cdot) - \pi\|_{\mathsf{TV}}$

Rapidly mixing Markov chains

Markov chain (X_t), state space $\mathcal{X}$, transition prob. $P : \mathcal{X} \times \mathcal{X} \to [0,1]$

- If *irreducible* and *aperiodic*: Converges to *stationary distribution* π:

$$\mathbb{P}[X_t = x] \to \pi(x)$$

- If P symmetric: π equals uniform distribution.
- *Total variation distance* of distributions μ, μ' on $\mathcal{X}$:

$$\|\mu - \mu'\|_{\mathsf{TV}} := \sup_{M \subseteq \mathcal{X}} |\mu(M) - \mu'(M)|$$

- Define $d(t) := \max_{x \in \mathcal{X}} \|P^t(x, \cdot) - \pi\|_{\mathsf{TV}}$
- *Mixing time* $\tau(\varepsilon) := \min\{t \in \mathbb{N}_0 : \forall t' \geq t : d(t') \leq \varepsilon\}$

Rapidly mixing Markov chains

Markov chain (X_t), state space $\mathcal{X}$, transition prob. $P : \mathcal{X} \times \mathcal{X} \to [0, 1]$

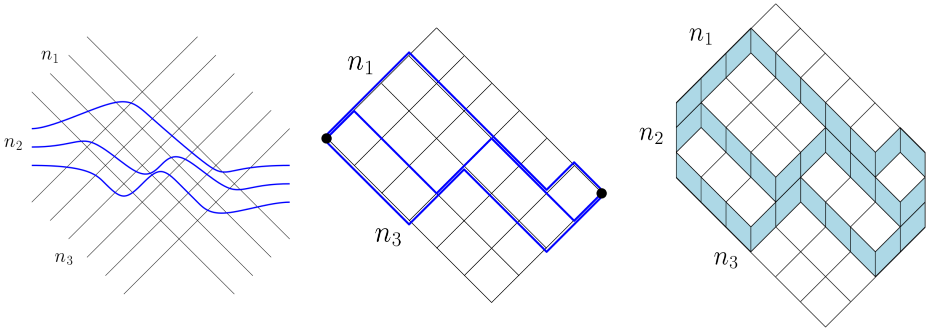
- If *irreducible* and *aperiodic*: Converges to *stationary distribution* π:
$$\mathbb{P}[X_t = x] \to \pi(x)$$
- If P symmetric: π equals uniform distribution.
- *Total variation distance* of distributions μ, μ' on $\mathcal{X}$:
$$\|\mu - \mu'\|_{\mathsf{TV}} := \sup_{M \subseteq \mathcal{X}} |\mu(M) - \mu'(M)|$$
- Define $d(t) := \max_{x \in \mathcal{X}} \|P^t(x, \cdot) - \pi\|_{\mathsf{TV}}$
- *Mixing time* $\tau(\varepsilon) := \min\{t \in \mathbb{N}_0 : \forall t' \geq t : d(t') \leq \varepsilon\}$

Definition

A class of Markov chains is *rapidly mixing* if for each of them $\tau(\varepsilon) \in \mathcal{O}\left(p\left(\log \frac{|\mathcal{X}|}{\varepsilon}\right)\right)$ for some $p \in \mathbb{R}[X]$.

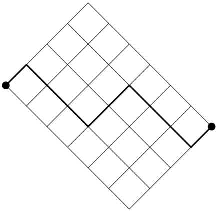
Pseudoline arrangements with 3 parallel classes

- **Ruby, Randall, Sinclair, 2001:**
  Rapidly mixing Marcov chain for sampling PLA with 3 parallel classes
- Idea: Extension of grid along non-crossing monotonic paths

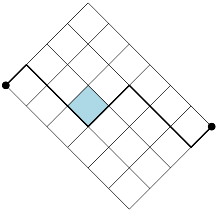


- On non-crossing path systems perform flips over grid cells.

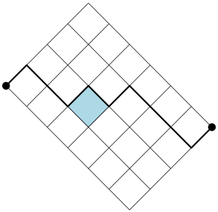
- Simple case: Only one insertion path



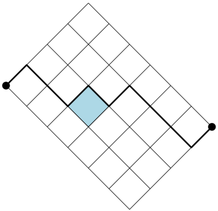
- Simple case: Only one insertion path



- Simple case: Only one insertion path

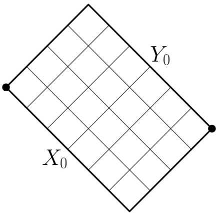


- Simple case: Only one insertion path



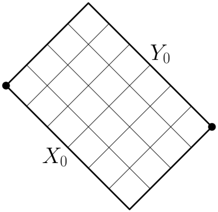
- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.

- Simple case: Only one insertion path



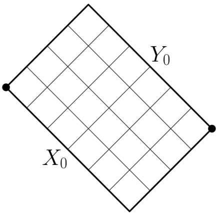
- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.
  - X_0 lowest path, Y_0 highest path, $X_0 \leq Y_0$

- Simple case: Only one insertion path



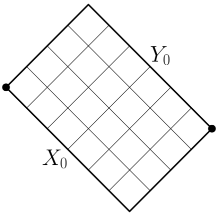
- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.
  - X_0 lowest path, Y_0 highest path, $X_0 \leq Y_0$
  - Choose (X_{t+1}, Y_{t+1}) from (X_t, Y_t) by choosing same flip.

- Simple case: Only one insertion path



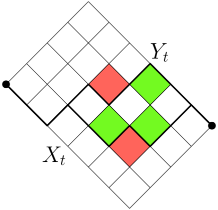
- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.
  - X_0 lowest path, Y_0 highest path, $X_0 \leq Y_0$
  - Choose (X_{t+1}, Y_{t+1}) from (X_t, Y_t) by choosing same flip.
  - Preserves $X_t \leq Y_t$ for all t.

- Simple case: Only one insertion path



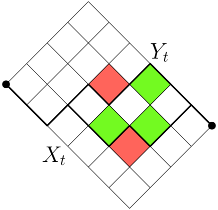
- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.
  - X_0 lowest path, Y_0 highest path, $X_0 \leq Y_0$
  - Choose (X_{t+1}, Y_{t+1}) from (X_t, Y_t) by choosing same flip.
  - Preserves $X_t \leq Y_t$ for all t.
  - Random variable $\tau_C := \min\{t : X_t = Y_t\}$

- Simple case: Only one insertion path



- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.
  - X_0 lowest path, Y_0 highest path, $X_0 \leq Y_0$
  - Choose (X_{t+1}, Y_{t+1}) from (X_t, Y_t) by choosing same flip.
  - Preserves $X_t \leq Y_t$ for all t.
  - Random variable $\tau_C := \min\{t : X_t = Y_t\}$
- Get polynomial bound on $\mathbb{E}[\tau_C]$ by upper bounding expected change of area between X_t and Y_t: $\mathbb{E}[\triangle d(X_t, Y_t)] \leq 0$.

- Simple case: Only one insertion path



- **Technique:** *Monotone coupling*
  - Partial order on paths: $p \leq q$ iff p below q.
  - X_0 lowest path, Y_0 highest path, $X_0 \leq Y_0$
  - Choose (X_{t+1}, Y_{t+1}) from (X_t, Y_t) by choosing same flip.
  - Preserves $X_t \leq Y_t$ for all t.
  - Random variable $\tau_C := \min\{t : X_t = Y_t\}$
- Get polynomial bound on $\mathbb{E}[\tau_C]$ by upper bounding expected change of area between X_t and Y_t: $\mathbb{E}[\triangle d(X_t, Y_t)] \leq 0$.
- From theory: $\tau(\varepsilon) \leq 6 \cdot \mathbb{E}[\tau_C] \left(1 + \log\left(\frac{1}{\varepsilon}\right)\right)$

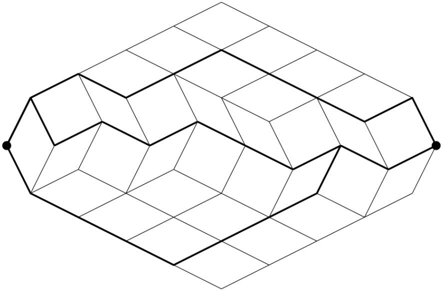
Insertion of new parallel class

- **Given:** Pseudoline arrangement with r parallel classes
- **Task:** Insert $(r+1)$-th parallel class with n_{r+1} pseudolines (uniformly by random among all possibilities)

Insertion of new parallel class

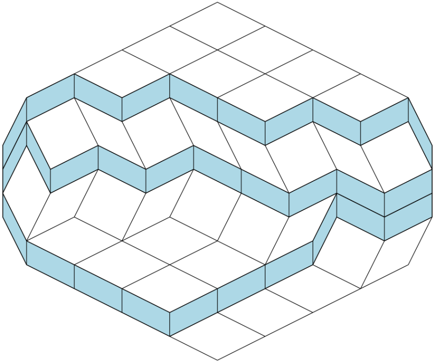
- **Given:** Pseudoline arrangement with r parallel classes
- **Task:** Insert $(r+1)$-th parallel class with n_{r+1} pseudolines (uniformly by random among all possibilities)
- Have seen this for $r = 2$, for $r > 2$?

- **Given:** Pseudoline arrangement with r parallel classes
- **Task:** Insert $(r+1)$-th parallel class with n_{r+1} pseudolines (uniformly by random among all possibilities)
- Have seen this for $r = 2$, for $r > 2$?
- **Same idea:** Markov chain with flips on non-crossing mon. paths

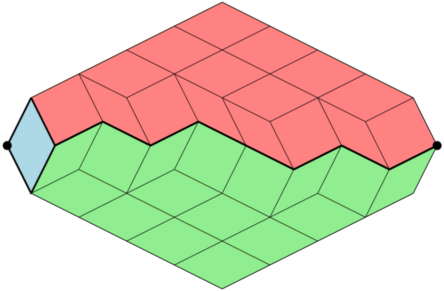


Insertion of new parallel class

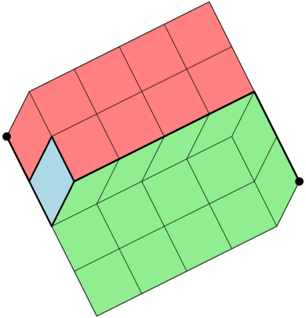
- **Given:** Pseudoline arrangement with r parallel classes
- **Task:** Insert $(r+1)$-th parallel class with n_{r+1} pseudolines (uniformly by random among all possibilities)
- Have seen this for $r = 2$, for $r > 2$?
- **Same idea:** Markov chain with flips on non-crossing mon. paths



Bottlenecks in path insertion chain



Extension of $r = 4$ classes

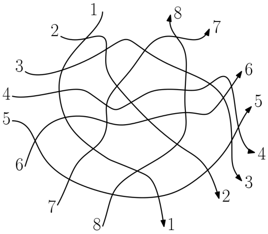


Extension of $r = 3$ classes

- Partition of paths into two classes:
  - A: Paths **above blue cell**, never going through green cells
  - $A^{\complement}$: Paths **below blue cell**, never going through red cells
- Flip on blue cell is only transition between A and $A^{\complement}$.
- **Result:** Path insertion Markov chain not rapidly mixing for $r \geq 3$.

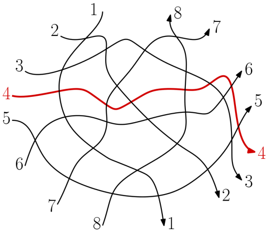
Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:



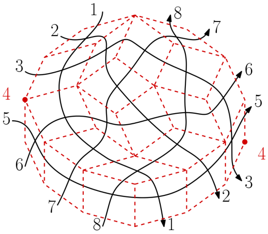
Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:



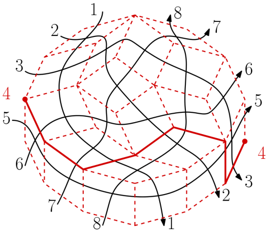
Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:



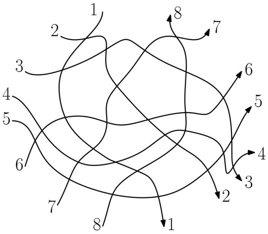
Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:



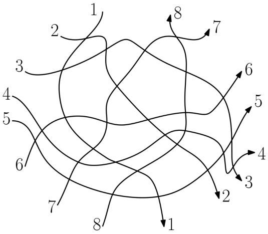
Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:



Repeated reinsertion of pseudolines

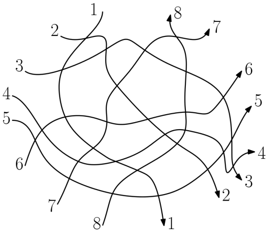
- Reinsertion of randomly chosen pseudoline defines Markov chain:



- Rapidly mixing?

Repeated reinsertion of pseudolines

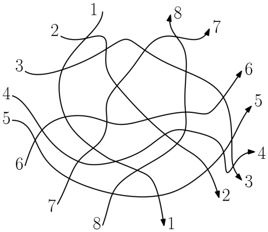
- Reinsertion of randomly chosen pseudoline defines Markov chain:



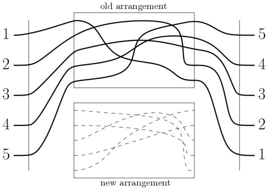
- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

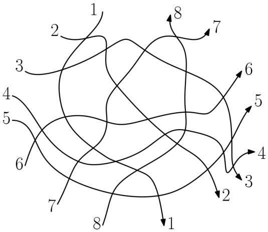


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

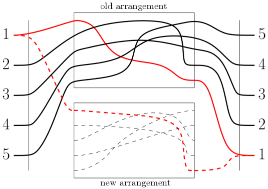


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

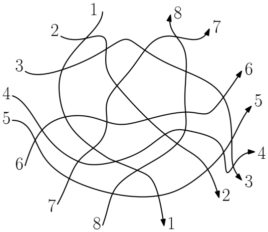


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

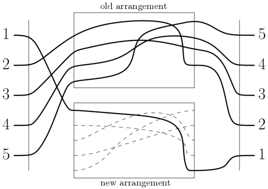


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

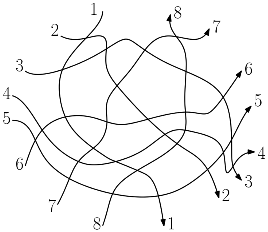


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

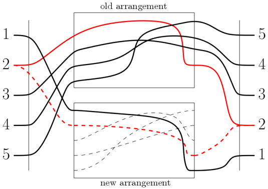


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

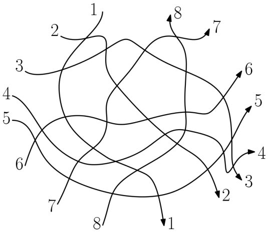


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

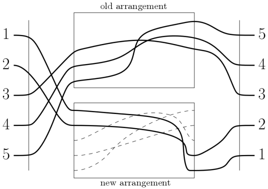


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

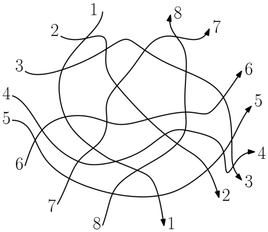


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

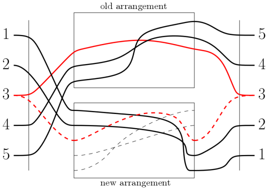


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

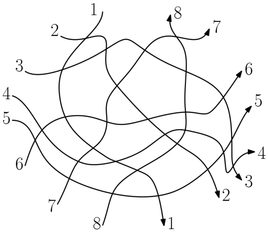


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

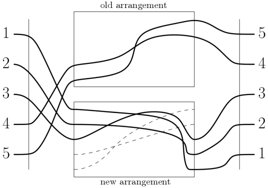


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

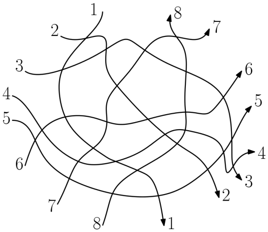


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

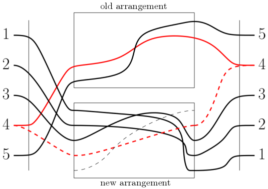


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:

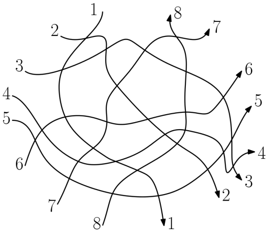


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

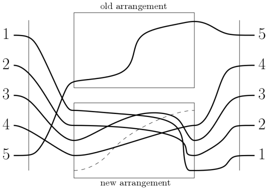


Repeated reinsertion of pseudolines

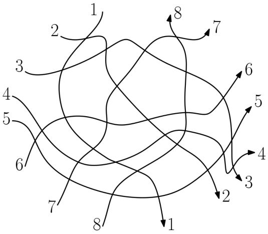
- Reinsertion of randomly chosen pseudoline defines Markov chain:



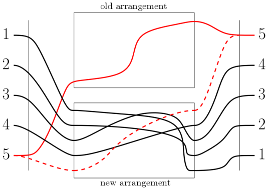
- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:



- Reinsertion of randomly chosen pseudoline defines Markov chain:

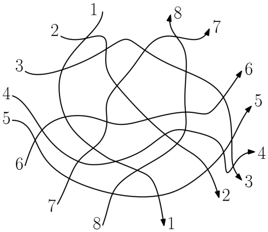


- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:

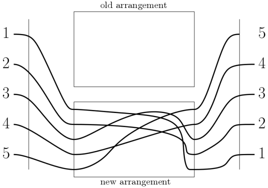


Repeated reinsertion of pseudolines

- Reinsertion of randomly chosen pseudoline defines Markov chain:



- Rapidly mixing?
- From every PLA to every PLA by touching each pseudline once:



Canonical paths

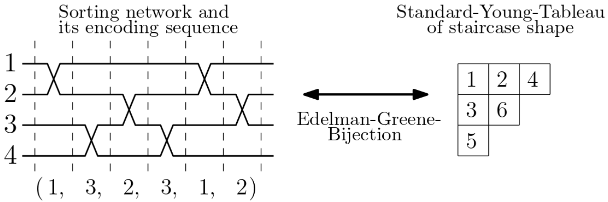
- For all $s, t \in \mathcal{X}$ find *canonical path* γ_{st} of transitions from s to t.
- Define *path congestion*:

$$\rho := \max_{\substack{x,y \in \mathcal{X} \\ P(x,y) \neq 0}} \frac{1}{\pi(x) P(x,y)} \sum_{\gamma_{st} \ni (x,y)} \pi(s)\pi(t)$$

- Low path congestion $\rho \Rightarrow$ No bottlenecks $\Rightarrow$ Low mixing time $\tau(\varepsilon)$
- Could be useful tool for „Reinsertion markov chain“.

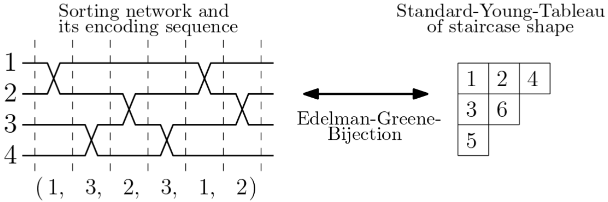
Rejection sampling via Standard-Young-Tableaux

- PLAs of shape $(n_1, ..., n_r) = (1, ..., 1)$ as *Sorting networks*:



Rejection sampling via Standard-Young-Tableaux

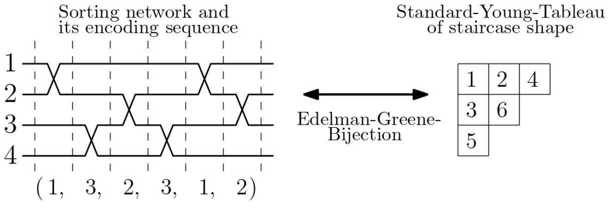
- PLAs of shape $(n_1, ..., n_r) = (1, ..., 1)$ as *Sorting networks*:



- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)

Rejection sampling via Standard-Young-Tableaux

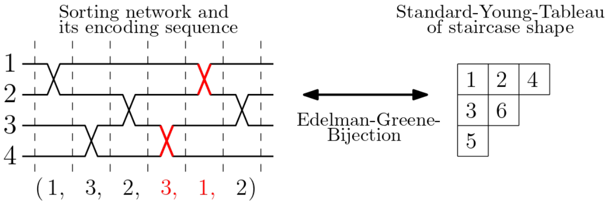
- PLAs of shape $(n_1, ..., n_r) = (1, ..., 1)$ as *Sorting networks*:



- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks

Rejection sampling via Standard-Young-Tableaux

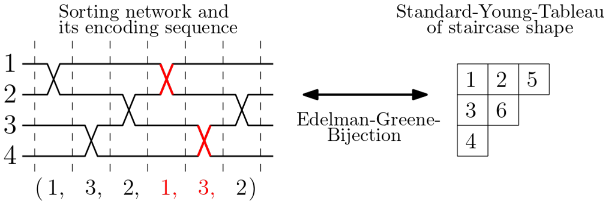
- PLAs of shape $(n_1, ..., n_r) = (1, ..., 1)$ as *Sorting networks*:



- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks

Rejection sampling via Standard-Young-Tableaux

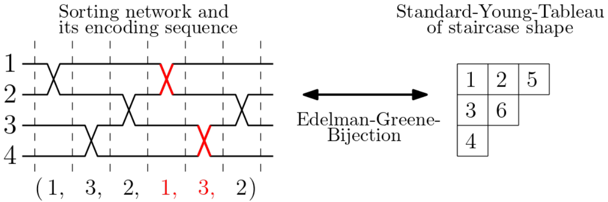
- PLAs of shape $(n_1, ..., n_r) = (1, ..., 1)$ as *Sorting networks*:



- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks

Rejection sampling via Standard-Young-Tableaux

- PLAs of shape $(n_1, \dots, n_r) = (1, \dots, 1)$ as *Sorting networks*:

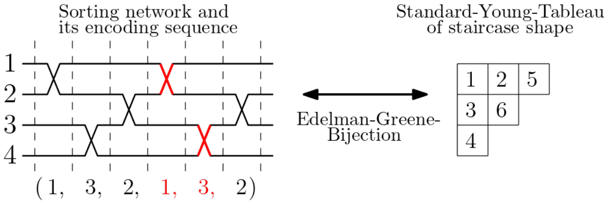


- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks
- If $|s_i - s_{i+1}| \geq 2$, then

$$(s_1, \dots, s_i, s_{i+1}, \dots, s_{\binom{r}{2}}) \sim (s_1, \dots, s_{i+1}, s_i, \dots, s_{\binom{r}{2}})$$

Rejection sampling via Standard-Young-Tableaux

- PLAs of shape $(n_1, \dots, n_r) = (1, \dots, 1)$ as *Sorting networks*:



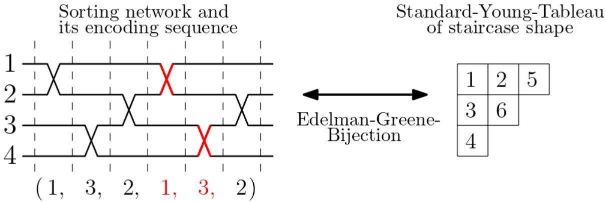
- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks
- If $|s_i - s_{i+1}| \geq 2$, then

$$(s_1, \dots, s_i, s_{i+1}, \dots, s_{\binom{r}{2}}) \sim (s_1, \dots, s_{i+1}, s_i, \dots, s_{\binom{r}{2}})$$

- **Def:** Sequence $(s_1, \dots, s_{\binom{r}{2}})$ *left extreme* iff $\nexists (s_i, s_{i+1}) : s_i \geq s_{i+1} + 2$.

Rejection sampling via Standard-Young-Tableaux

- PLAs of shape $(n_1, \dots, n_r) = (1, \dots, 1)$ as *Sorting networks*:



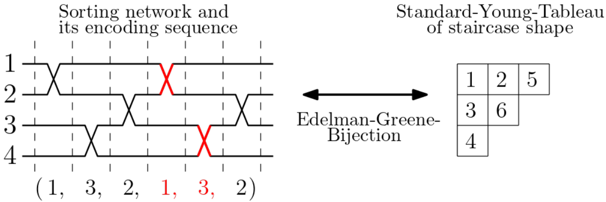
- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks
- If $|s_i - s_{i+1}| \geq 2$, then

$$(s_1, \dots, s_i, s_{i+1}, \dots, s_{\binom{r}{2}}) \sim (s_1, \dots, s_{i+1}, s_i, \dots, s_{\binom{r}{2}})$$

- **Def:** Sequence $(s_1, \dots, s_{\binom{r}{2}})$ *left extreme* iff $\nexists (s_i, s_{i+1}) : s_i \geq s_{i+1} + 2$.
- **Idea:** Sample SYT and accept PLA if sequence is left extreme.

Rejection sampling via Standard-Young-Tableaux

- PLAs of shape $(n_1, \ldots, n_r) = (1, \ldots, 1)$ as *Sorting networks*:



- Efficient sampling of Std. Young Tableaux (SYT) possible. (**Greene, Nijenhuis, Wilf, 1979**)
- **No direct bijection!** PLAs = Equivalence classes of Sorting networks
- If $|s_i - s_{i+1}| \geq 2$, then

$$(s_1, \ldots, s_i, s_{i+1}, \ldots, s_{\binom{r}{2}}) \sim (s_1, \ldots, s_{i+1}, s_i, \ldots, s_{\binom{r}{2}})$$

- **Def:** Sequence $(s_1, \ldots, s_{\binom{r}{2}})$ *left extreme* iff $\nexists (s_i, s_{i+1}) : s_i \geq s_{i+1} + 2$.
- **Idea:** Sample SYT and accept PLA if sequence is left extreme.
- **Experimental result:** Sampling PLAs with $r \leq 8$ in reasonable time

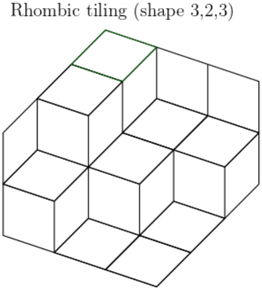
Optimal plane partitions

- **Def:** Matrix $[h_{i,j}] \in {\mathbb{N}_0}^{r \times s}$ is called *plane partition*, if rows and columns are monotonically increasing.

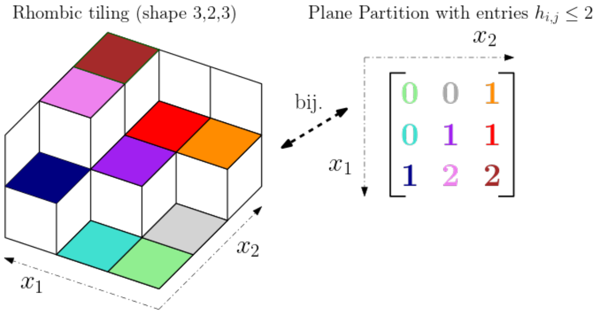
- **Def:** Matrix $[h_{i,j}] \in {\mathbb{N}_0}^{r \times s}$ is called *plane partition*, if rows and columns are monotonically increasing.
- For objective functions $f_{i,j} : \{0, ..., h\} \to \mathbb{Q}$ consider the problem

$$
\begin{aligned}
\textbf{min} \quad & \sum_{i=1}^{r} \sum_{j=1}^{s} f_{i,j}(h_{i,j}) & \\
\text{s.t.} \quad & h_{i,j} \leq h_{i,j+1} & \text{für alle } i \in [r], j \in [s-1] \\
& h_{i,j} \leq h_{i+1,j} & \text{für alle } i \in [r-1], j \in [s] \\
& 0 \leq h_{i,j} \leq h & \\
& h_{i,j} \in \mathbb{Z} &
\end{aligned}
$$

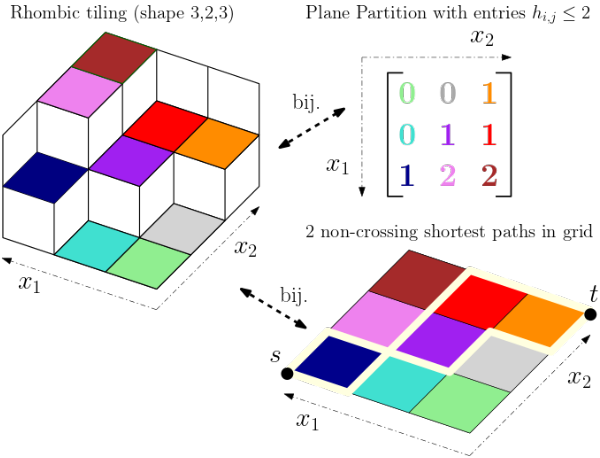
- *Plane partitions*, *rhombic tilings* and *systems of non-crossing paths* in bijection:



- *Plane partitions*, *rhombic tilings* and *systems of non-crossing paths* in bijection:



- *Plane partitions*, *rhombic tilings* and *systems of non-crossing paths* in bijection:



- **Equivalent problem:** On set $\mathcal{P}$ of systems of h non-crossing monotonic paths in $(r \times s)$-grid graph:

$$\begin{aligned} \min \quad & \sum_{k=1}^{h} \sum_{e \in p_k} w_k(e) \\ \text{s.t.} \quad & (p_1, \ldots, p_h) \in \mathcal{P} \end{aligned}$$

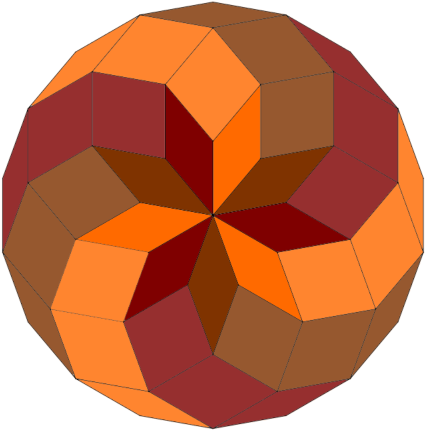
where $w_k : E \to \mathbb{Q}$ some weights determined by $f_{i,j}$.

- **Equivalent problem:** On set $\mathcal{P}$ of systems of h non-crossing monotonic paths in $(r \times s)$-grid graph:

$$\begin{aligned} \min \quad & \sum_{k=1}^{h} \sum_{e \in p_k} w_k(e) \\ \text{s.t.} \quad & (p_1, \dots, p_h) \in \mathcal{P} \end{aligned}$$

  where $w_k : E \to \mathbb{Q}$ some weights determined by $f_{i,j}$.
- Can be solved by dynamic programming approach in time polynomial in r, s but exponential in h.

Questions



Exact references in submitted thesis or on demand.