The Tree Growing Sequence

Carrie Frizzell

Kansas State University

AMS Central Sectional Meeting
September 12-13, 2020

Basics

Notation

- Work with $G = (V, E)$ a multigraph. Order any multiple edges between two vertices. Choose a root vertex and call it q.

- Work with $G = (V, E)$ a multigraph. Order any multiple edges between two vertices. Choose a root vertex and call it q.
- $G - e$: delete the edge e
- G/e: contract the edge e

Basics

Notation

- $\mathcal{P}_{G,q}$ - set of G-parking functions with respect to q
- $\mathcal{T}_G$ - set of spanning trees
- $\mathcal{M}_G$ - multiset of monomials of $T(G; x, y)$, the Tutte polynomial

Background

$$
\begin{tikzcd}
 & \mathcal{P}_{G,q} \arrow[dl, leftrightarrow] \arrow[dd, leftrightarrow] \\
\mathcal{T}_G \arrow[dr, leftrightarrow] & \\
 & \mathcal{M}_G
\end{tikzcd}
$$

- Dhar [Dha90]
- Biggs [Big99]
- Cori and Le Borgne [CB03]
- Chebikin and Pylyavskyy [CP05]
- Bernardi [Ber08]
- Kostic and Yan [KY08]
- Baker and Shokrieh [BS13]

Many more

Definition

Tutte Polynomial

$$T(G; x, y) = \sum_{\mathcal{T}_G} x^{ia} y^{ea}$$

Definition

Tutte Polynomial

$$T(G;x,y) = \sum_{\mathcal{T}_G} x^{ia} y^{ea}$$

(+ other closed formulas)

Definition

Tutte Polynomial

$$T(G;x,y) = \sum_{\mathcal{T}_G} x^{ia} y^{ea}$$

(+ other closed formulas)

$$T(G;x,y) = \begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

Definition

G-Parking Functions

The **outdegree with respect to** $A \subseteq V$, denoted $outdeg_A(v)$, is the number of edges from $v \in A$ to vertices not in A.

Definition

G-Parking Functions

The **outdegree with respect to** $A \subseteq V$, denoted $outdeg_A(v)$, is the number of edges from $v \in A$ to vertices not in A.

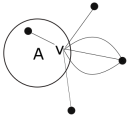
A G-**parking function** is a function $f : V(G) - \{q\} \to \mathbb{Z}_{\geq 0}$ such that any subset $A \subseteq V - \{q\}$ contains a vertex v with $0 \leq f(v) < outdeg_A(v)$.

Definition

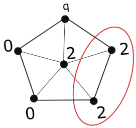
G-Parking Functions

The **outdegree with respect to** $A \subseteq V$, denoted $outdeg_A(v)$, is the number of edges from $v \in A$ to vertices not in A.

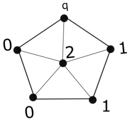
A G-**parking function** is a function $f : V(G) - \{q\} \to \mathbb{Z}_{\geq 0}$ such that any subset $A \subseteq V - \{q\}$ contains a vertex v with $0 \leq f(v) < outdeg_A(v)$.



outdegree$_A$(v)=5



Not G-parking



G-parking

Algorithms which produce bijective correspondences

- Dependence on choices

Algorithms which produce bijective correspondences

- Dependence on choices
  - Global edge order.
  - Vertex order.
  - Something else.

Algorithms which produce bijective correspondences

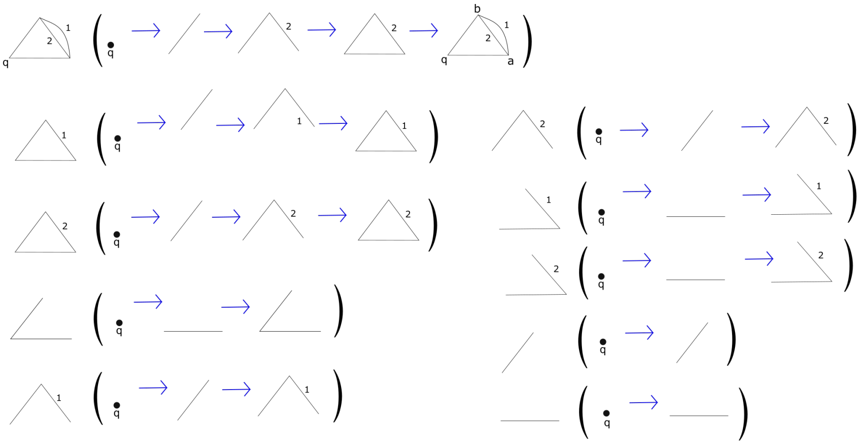
- Dependence on choices
  - Global edge order.
  - Vertex order.
  - Something else.
- Compatibility between these choices.
  - In general, no.
  - Would like to get all bijections in the triangle using one algorithm, but do it in a non-arbitrary way.

TGS - Definition

Definition: Given a connected graph $G = (V, E)$ and the set $\mathcal{S}$ of all subgraphs of G containing q as a vertex, a **tree growing sequence (TGS)** is a collection of tuples

$$\Sigma = \{(S, \sigma_S : \ \mathcal{H}_S \to \ E(S))\}$$

where $S \in \mathcal{S}$, σ_S is a function on the set $\mathcal{H}_S$ of "rooted" subgraphs of S, $\sigma_S(T) \notin E(T)$, and $\sigma_S(T) \cup T$ is connected.


q
1
2
b
a

TGS - Algorithm

Given a TGS $\Sigma = \{(S, \sigma_S)\}$ and $f : V - \{q\} \to \mathbb{Z}$.

TGS - Algorithm

Given a TGS $\Sigma = \{(S, \sigma_S)\}$ and $f : V - \{q\} \to \mathbb{Z}$.
Input: $(f, S, U, X, \alpha, \beta)$

TGS - Algorithm

Given a TGS $\Sigma = \{(S, \sigma_S)\}$ and $f : V - \{q\} \to \mathbb{Z}$.

Input: $(f, S, U, X, \alpha, \beta)$

Output: A tree T_f and monomial $x^\alpha y^\beta$

TGS - Algorithm

Given a TGS $\Sigma = \{(S, \sigma_S)\}$ and $f : V - \{q\} \to \mathbb{Z}$.

Input: $(f, S, U, X, \alpha, \beta)$

Output: A tree T_f and monomial $x^\alpha y^\beta$

Initially: $S = G, U = \{q\}$, $X = \{\emptyset\} \subset E(G)$, $\alpha = 0$, $\beta = 0$.

TGS - Algorithm

Given a TGS $\Sigma = \{(S, \sigma_S)\}$ and $f : V - \{q\} \to \mathbb{Z}$.

Input: $(f, S, U, X, \alpha, \beta)$

Output: A tree T_f and monomial $x^\alpha y^\beta$

Initially: $S = G, U = \{q\}$, $X = \{\emptyset\} \subset E(G)$, $\alpha = 0$, $\beta = 0$.

At each step, consider $e = \sigma_S(T)$, where $T = (U, X)$. Let $e = (w, v)$, where $w \in U$. We care about $f(v)$ and the nature of e.

TGS - Algorithm

Given a TGS $\Sigma = \{(S, \sigma_S)\}$ and $f : V - \{q\} \to \mathbb{Z}$.

Input: $(f, S, U, X, \alpha, \beta)$

Output: A tree T_f and monomial $x^\alpha y^\beta$

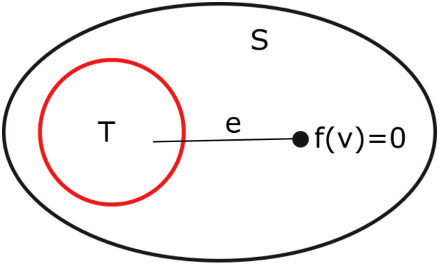
Initially: $S = G, U = \{q\}$, $X = \{\emptyset\} \subset E(G)$, $\alpha = 0$, $\beta = 0$.

At each step, consider $e = \sigma_S(T)$, where $T = (U, X)$. Let $e = (w, v)$, where $w \in U$. We care about $f(v)$ and the nature of e.

If $f(v) < 0$, delete e and terminate the algorithm.

TGS - Algorithm

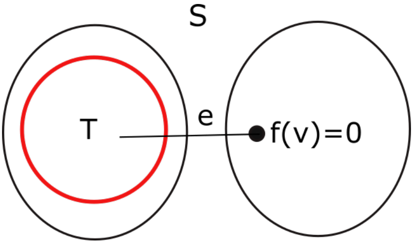
$$(f, S, U, X, \alpha, \beta) \longrightarrow (f, S, U \cup v, X \cup e, \alpha, \beta)$$



Next step: $\sigma_S(T)$.

TGS - Algorithm

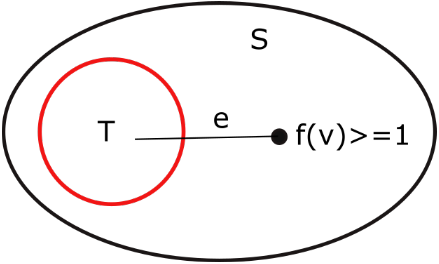
$$(f, S, U, X, \alpha, \beta) \longrightarrow (f, S, U \cup v, X \cup e, \alpha + 1, \beta)$$



Next: $\sigma_S(T)$

TGS - Algorithm

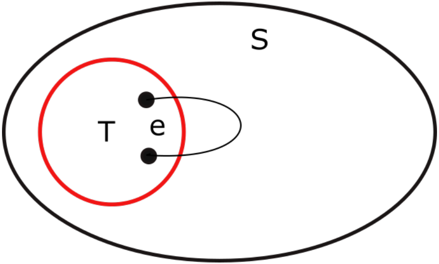
$$(f, S, U, X, \alpha, \beta) \longrightarrow (f, S - e, U, X, \alpha, \beta)$$



Next: $\sigma_{S-e}(T)$.

TGS - Algorithm

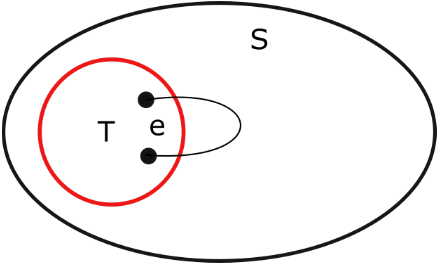
$$(f, S, U, X, \alpha, \beta) \longrightarrow (f, S - e, U, X, \alpha, \beta + 1)$$



Next: $\sigma_{S-e}(T)$.

TGS - Algorithm

$$(f, S, U, X, \alpha, \beta) \longrightarrow (f, S - e, U, X, \alpha, \beta + 1)$$



Next: $\sigma_{S-e}(T)$.

Terminate when $T_f = (U, X)$ spans the connected component of S containing q.

TGS - Splitting

Start with $e = (q, v)$.

$$\{f \in \mathcal{P}_{G,q} \mid f(v) = 0\} \longleftrightarrow \{\mathcal{P}_{G/e,q}\}$$

$$\{f \in \mathcal{P}_{G,q} \mid f(v) \geq 1\} \longleftrightarrow \{\mathcal{P}_{G-e,q}\}$$

TGS - Splitting

Start with $e = (q, v)$.

$$\{f \in \mathcal{P}_{G,q} \,|\, f(v) = 0\} \longleftrightarrow \{\mathcal{P}_{G/e,q}\}$$

$$\{f \in \mathcal{P}_{G,q} \,|\, f(v) \geq 1\} \longleftrightarrow \{\mathcal{P}_{G-e,q}\}$$

It follows that for any subgraph S and any edge e in S, $\mathcal{P}_{S/e} \sqcup \mathcal{P}_{S-e}$ is in $1-1$ correspondence with $P_{S,q}$.

TGS - Splitting

Splitting based on the recursive formula for the Tutte polynomial.

$$T(G;x,y) = \begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

TGS - Splitting

Splitting based on the recursive formula for the Tutte polynomial.

$$T(G;x,y) = \begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

$$\mathcal{M}_G = \mathcal{M}_{G/e} \sqcup \mathcal{M}_{G-e}, \text{ or}$$

TGS - Splitting

Splitting based on the recursive formula for the Tutte polynomial.

$$T(G;x,y) = \begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

$$\mathcal{M}_G = \mathcal{M}_{G/e} \sqcup \mathcal{M}_{G-e}, \text{ or}$$

$$\mathcal{M}_G = x \cdot \mathcal{M}_{G/e}, \text{ or}$$

TGS - Splitting

Splitting based on the recursive formula for the Tutte polynomial.

$$T(G;x,y) = \begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

$$\mathcal{M}_G = \mathcal{M}_{G/e} \sqcup \mathcal{M}_{G-e}, \text{ or}$$

$$\mathcal{M}_G = x \cdot \mathcal{M}_{G/e}, \text{ or}$$

$$\mathcal{M}_G = y \cdot \mathcal{M}_{G-e}$$

TGS - Splitting

Splitting based on the recursive formula for the Tutte polynomial.

$$T(G;x,y) = \begin{cases} yT(G-e;x,y) & e \text{ a loop} \\ xT(G/e;x,y) & e \text{ a bridge} \\ T(G-e;x,y) + T(G/e;x,y) & \text{otherwise} \end{cases}$$

$$\mathcal{M}_G = \mathcal{M}_{G/e} \sqcup \mathcal{M}_{G-e}, \text{ or}$$

$$\mathcal{M}_G = x \cdot \mathcal{M}_{G/e}, \text{ or}$$

$$\mathcal{M}_G = y \cdot \mathcal{M}_{G-e}$$

Furthermore, we have $\mathcal{T}_G \leftrightarrow \mathcal{T}_{G/e} \sqcup \mathcal{T}_{G-e}$

TGS - Splitting

Proposition:

Let τ be the assignments $f \mapsto T_f$ and ρ be the assignments $f \mapsto x^{\alpha} y^{\beta}$. These maps are bijective.

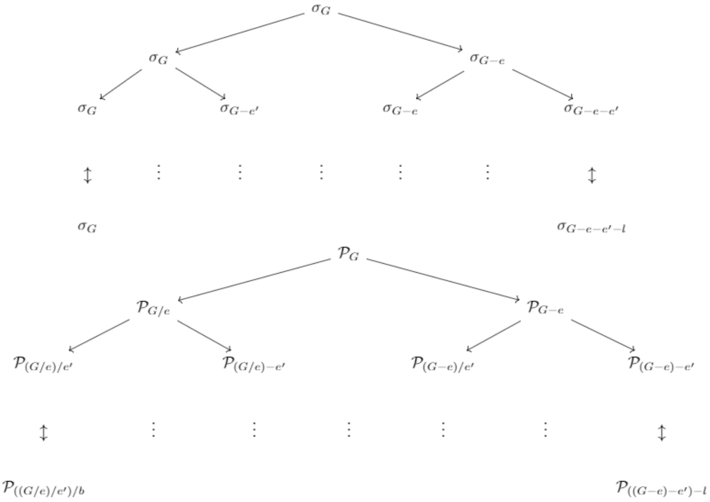
TGS - Splitting

Proposition:

Let τ be the assignments $f \mapsto T_f$ and ρ be the assignments $f \mapsto x^\alpha y^\beta$. These maps are bijective.

$$\begin{tikzcd}[column sep=small] & \mathcal{P}_{G,q}, \mathcal{T}_G, \mathcal{M}_G \arrow[dl] \arrow[dr] & \\ \mathcal{P}_{G/e,q}, \mathcal{T}_{G/e}, \mathcal{M}_{G/e} & & \mathcal{P}_{G-e,q}, \mathcal{T}_{G-e}, \mathcal{M}_{G-e} \end{tikzcd}$$

TGS - Splitting



Compare with Established Algorithms

Dhar's Burning Algorithm

Start with something simple.

Let O_E be a global edge order, D the application of Dhar's algorithm to $\mathcal{P}_{G,q}$, and K the composition of D with the bijection $\mathcal{T}_G \to \mathcal{M}_G$ arising from internal and external activities.

Compare with Established Algorithms

Dhar's Burning Algorithm

Start with something simple.

Let O_E be a global edge order, D the application of Dhar's algorithm to $\mathcal{P}_{G,q}$, and K the composition of D with the bijection $\mathcal{T}_G \to \mathcal{M}_G$ arising from internal and external activities.

Proposition: The diagrams commute.

$$
\begin{tikzcd}
\{O_E\} \arrow[r, "R"] \arrow[dr, "D"'] & \{\Sigma\} \arrow[d, "F"] \\
& \{\tau\},
\end{tikzcd}
\qquad
\begin{tikzcd}
\{O_E\} \arrow[r, "R"] \arrow[dr, "K"'] & \{\Sigma\} \arrow[d, "F"] \\
& \{\rho\}
\end{tikzcd}
$$

Compare with Established Algorithms

Family of Chebikin and Pylyavskyy [CP05]

Something more interesting.

Proper set of tree orders: Given an ordering $\pi(T)$ on the vertices of every subtree T rooted at q, the collection $\Pi_G = \{\pi(T) \mid T \subset G \text{ a rooted tree}\}$ is a proper set of tree orders if

Compare with Established Algorithms

Family of Chebikin and Pylyavskyy [CP05]

Something more interesting.

Proper set of tree orders: Given an ordering $\pi(T)$ on the vertices of every subtree T rooted at q, the collection $\Pi_G = \{\pi(T) \mid T \subset G \text{ a rooted tree}\}$ is a proper set of tree orders if

- When the overlap of T and T' contains a rooted tree, and i, j are vertices in this overlap, then $i <_{\pi(T)} j \iff i <_{\pi(T')} j$

Compare with Established Algorithms

Family of Chebikin and Pylyavskyy [CP05]

Something more interesting.

Proper set of tree orders: Given an ordering $\pi(T)$ on the vertices of every subtree T rooted at q, the collection $\Pi_G = \{\pi(T) \mid T \subset G \text{ a rooted tree}\}$ is a proper set of tree orders if

- When the overlap of T and T' contains a rooted tree, and i, j are vertices in this overlap, then $i <_{\pi(T)} j \iff i <_{\pi(T')} j$
- A directed edge $(u, v) \in T$ means $v < u$ in $\pi(T)$.

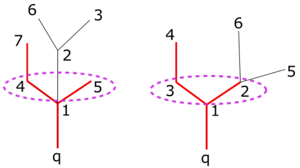
Compare with Established Algorithms

Family of Chebikin and Pylyavskyy [CP05]

Something more interesting.

Proper set of tree orders: Given an ordering $\pi(T)$ on the vertices of every subtree T rooted at q, the collection $\Pi_G = \{\pi(T) \mid T \subset G \text{ a rooted tree}\}$ is a proper set of tree orders if

- When the overlap of T and T' contains a rooted tree, and i, j are vertices in this overlap, then $i <_{\pi(T)} j \iff i <_{\pi(T')} j$
- A directed edge $(u, v) \in T$ means $v < u$ in $\pi(T)$.



Not Proper

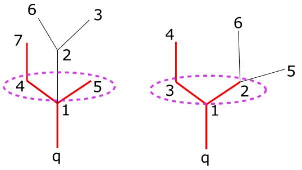
Compare with Established Algorithms

Family of Chebikin and Pylyavskyy [CP05]

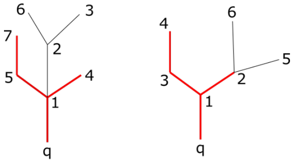
Something more interesting.

Proper set of tree orders: Given an ordering $\pi(T)$ on the vertices of every subtree T rooted at q, the collection $\Pi_G = \{\pi(T) \mid T \subset G \text{ a rooted tree}\}$ is a proper set of tree orders if

- When the overlap of T and T' contains a rooted tree, and i, j are vertices in this overlap, then $i <_{\pi(T)} j \iff i <_{\pi(T')} j$
- A directed edge $(u, v) \in T$ means $v < u$ in $\pi(T)$.

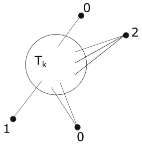


Not Proper



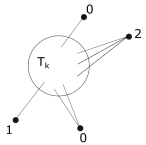
Proper

Bijection $\Phi : \mathcal{P}_{G,q} \to \mathcal{T}_G$ for every Π_G.

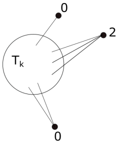


Consider all vertices incident to T_k.

Bijection $\Phi : \mathcal{P}_{G,q} \to \mathcal{T}_G$ for every Π_G.

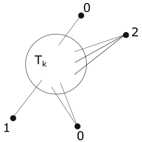


Consider all vertices incident to T_k.

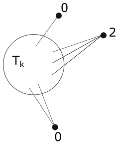


Keep v with f(v)<#edges to T_k.

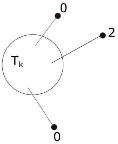
Bijection $\Phi : \mathcal{P}_{G,q} \to \mathcal{T}_G$ for every Π_G.



Consider all vertices incident to T_k.

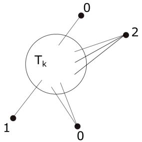


Keep v with f(v)<#edges to T_k.

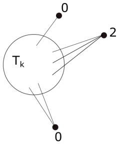


Keep the edge larger than f(v) other edges, according to order on T_k.

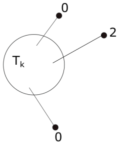
Bijection $\Phi : \mathcal{P}_{G,q} \to \mathcal{T}_G$ for every Π_G.



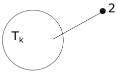
Consider all vertices incident to T_k.



Keep v with $f(v)<\#$edges to T_k.



Keep the edge larger than f(v) other edges, according to order on T_k.



Choose smallest edge according to order on $T_k \cup \{e\}$.

Compare with Established Algorithms

Family of Chebikin and Pylyavskyy

Define $\Omega : \{\Pi_G\} \to \{\Sigma\}$.

The TGS $\Omega(\Pi_G)$ may consider more edges at each step, and may add edges in a different order; however, the same spanning tree will be obtained as for Φ.

Compare with Established Algorithms

Family of Chebikin and Pylyavskyy

Define $\Omega : \{\Pi_G\} \to \{\Sigma\}$.

The TGS $\Omega(\Pi_G)$ may consider more edges at each step, and may add edges in a different order; however, the same spanning tree will be obtained as for Φ.

Proposition: *The map Ω is injective and the diagram commutes.*

$$\begin{tikzcd}
\{\Pi_G\} \arrow[r, "\Omega"] \arrow[dr, "\Phi"'] & \{\Sigma\} \arrow[d, "F"] \\
& \{\tau\}
\end{tikzcd}$$

Related Work by Other Authors

- Yuen
- Backman, Baker, Yuen
- Question posed by Hopkins: Classify tie-breaks for Dhar's algorithm.

Thank you!

Preprint: arXiv:2005.06456 (updates recently submitted)

References I

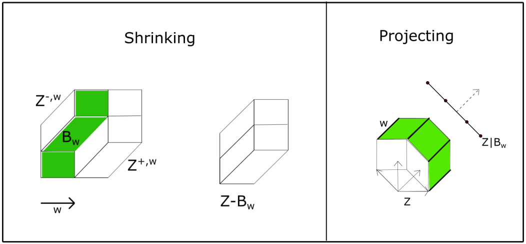
- Olivier Bernardi, *A characterization of the Tutte polynomial via combinatorial embeddings*, Annals of Combinatorics **12** (2008), no. 2, 139–153, 14 pages.
- N.L. Biggs, *Chip-firing and the critical group of a graph*, Journal of Algebraic Combinatorics **9** (1999), no. 1, 25–45.
- Matthew Baker and Farbod Shokrieh, *Chip-firing games, potential theory on graphs, and spanning trees*, Journal of Combinatorial Theory, Series A **120** (2013), no. 1, 164 – 182.
- Robert Cori and Yvan Le Borgne, *The sand-pile model and tutte polynomials*, Advances in Applied Mathematics **30** (2003), no. 1, 44 – 52.
- Denis Chebikin and Pavlo Pylyavskyy, *A family of bijections between g-parking functions and spanning trees*, Journal of Combinatorial Theory, Series A **110** (2005), no. 1, 31 – 41.

References II

Deepak Dhar, *Self-organized critical state of sandpile automaton models*, Phys. Rev. Lett. **64** (1990), 1613–1616.

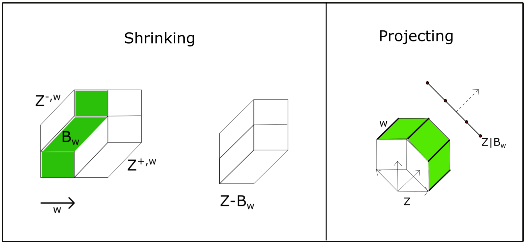
Dimitrije Kostić and Catherine H. Yan, *Multiparking functions, graph searching, and the tutte polynomial*, Advances in Applied Mathematics **40** (2008), no. 1, 73 – 97.

Tutte for a Zonotopal Tiling



$$T^*(\mathcal{Z};x,y) = \begin{cases} yT^*(\mathcal{Z} - B_w; x, y) & \mathcal{Z} - B_w \cong \mathcal{Z}|B_w \\ x^\gamma T^*(\mathcal{Z}|B_w; x, y) + T^*(\mathcal{Z} - B_w; x, y) & \text{otherwise} \end{cases}$$

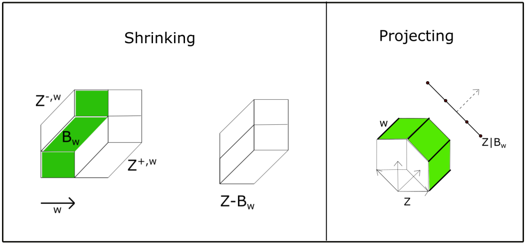
Tutte for a Zonotopal Tiling



$$T^*(\mathcal{Z};x,y) = \begin{cases} yT^*(\mathcal{Z}-B_w;x,y) & \mathcal{Z}-B_w \cong \mathcal{Z}|B_w \\ x^\gamma T^*(\mathcal{Z}|B_w;x,y) + T^*(\mathcal{Z}-B_w;x,y) & \text{otherwise} \end{cases}$$

- Visually, γ is the number of zones parallel to B_w; this is the same as the number of elements of the associated vector configuration $\mathcal{V}_\mathcal{Z}$ parallel to w (excluding w).

Tutte for a Zonotopal Tiling



$$T^*(\mathcal{Z};x,y) = \begin{cases} yT^*(\mathcal{Z}-B_w;x,y) & \mathcal{Z}-B_w \cong \mathcal{Z}|B_w \\ x^\gamma T^*(\mathcal{Z}|B_w;x,y) + T^*(\mathcal{Z}-B_w;x,y) & \text{otherwise} \end{cases}$$

- Visually, γ is the number of zones parallel to B_w; this is the same as the number of elements of the associated vector configuration $\mathcal{V}_\mathcal{Z}$ parallel to w (excluding w).
- Here, $\cong$ means as tiled zonotopes.

Zonotope - Cographical Matroid

Given a zonotopal tiling $\mathcal{Z}$, let $\mathcal{V}^*_{\mathcal{Z}}$ be the matroid with bases subsets of the configuration which are bases of $\mathbb{R}^d$. Thus, bases correspond to tiles. If the tiling arises from a graph, it is the *cographical matroid*.

Theorem: *Fix a cubical zonotopal tiling $\mathcal{Z}$ of Z with associated vector configuration $V_{\mathcal{Z}}$. Then $T^*(\mathcal{Z}; x, y)$ is the Tutte polynomial $T(\mathcal{V}^*_{\mathcal{Z}}; x, y)$.*

Further Questions

Question: *Are there choices which are compatible in that they make the diagram below commute?*

$$
\begin{tikzcd}
& \mathcal{P}_{G,q} \arrow[dl] \arrow[dr, leftrightarrow] \arrow[dd, leftrightarrow] & \\
\mathcal{M}_G & & \{\text{Tiles of } Z(G)\} \arrow[ll] \arrow[dl, leftrightarrow] \\
& \mathcal{T}_G \arrow[ul] &
\end{tikzcd}
$$