# Seminar WS 12/13: Graphs on Surfaces

# On the Number of Spanning Trees a Planar Graph can have

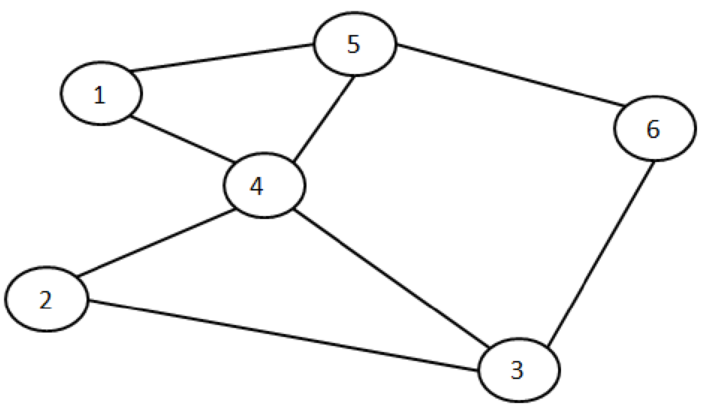
**Date:** December 6, 2012
**Presenter:** Stefan Bier
**Supervisor:** Dr. Mnich

# Prerequisites and Motivation I

**Definition :**
A **spanning tree** of a connected undirected Graph G = (V,E) is a minimal set of edges , that connect all vertices.
Equivalently it is a maximal set of edges, that contains no cycle.

❑

**Theorem (Kirchhoff's matrix tree theorem):**
The number of spanning trees of G is equal to the absolute value of *any* cofactor of the Laplacian matrix of G.




$$\begin{bmatrix} 2 & 0 & 0 & -1 & -1 & 0 \\ 0 & 2 & -1 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & -1 & 4 & -1 & 0 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

❑

# Prerequisites and Motivation II

- There are various settings where the number of spanning trees or the cofactors of the Laplacian play an important role.

- In our case the cofactors of the Laplacian are needed in conjunction with the embedding of **3D-polytopes** on preferably small integer grids.

- For this problem only the spanning trees of **planar graphs** are taken into consideration.

# Prerequisites and Motivation III

- We want to bound the number of spanning trees of **planar graphs from above.**
- From this we get bounds for the cofactors of the Laplacian and therefore bounds on the grid size of the 3D-polytope embedding.
- There are also effects on some related problems.
- Such bounds were known before, but the approach chosen by the authors leads to improved results.

# 3D-polytope integer grid embedding

**Definition :**
The **edge graph G(P) of a convex polytope P** is the connected graph whose vertex set is the vertex set of the polytope P, and two vertices are adjacent in the graph if they are endpoints of a 1-face of P.
❑

**Definition :**
A graph G=(V,E) is said to be **3-connected** if the graph remains connected when at most 2 vertices are deleted from the graph.
❑

**Why are only planar and 3-connected graphs of interest?**

# 3D-polytope integer grid embedding (continued)

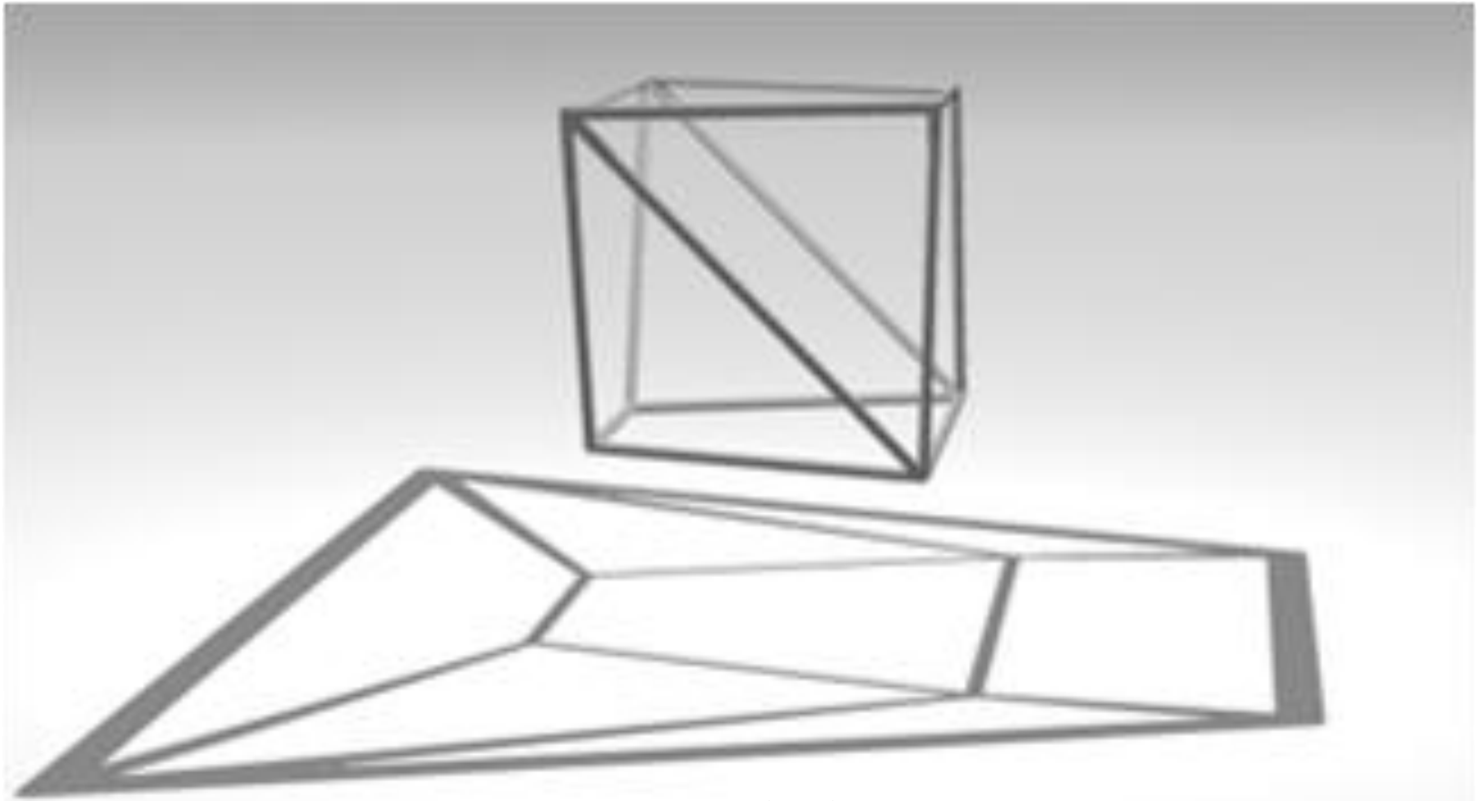
**Why are only planar and 3-connected graphs of interest?**

**Theorem (Steinitz):**
**A graph G is the edge graph of a convex 3D-polytope if and only if G is simple (no loops, no multiple edges), planar and 3-connected.**

❑

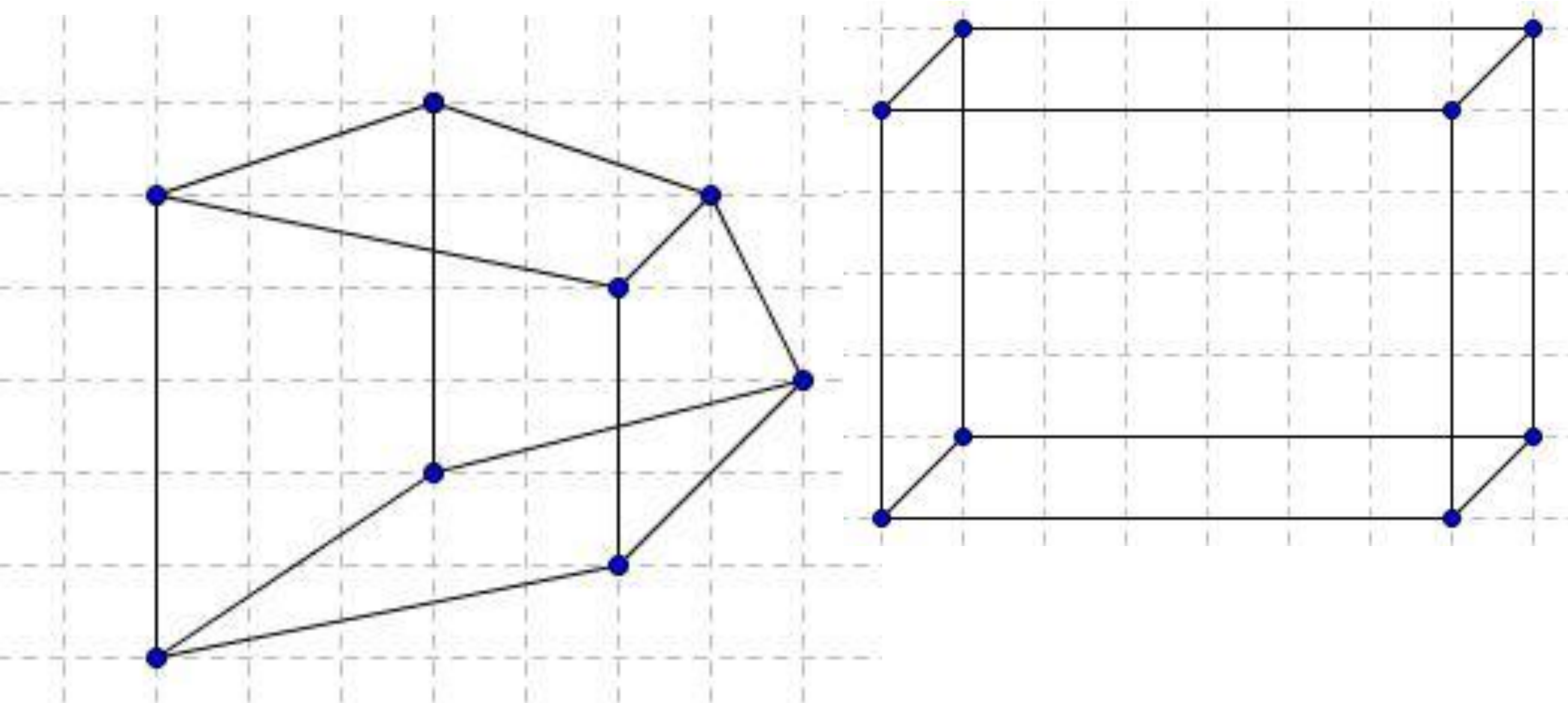
(All graphs in this context are simple.)

# Steinitz: Intuition



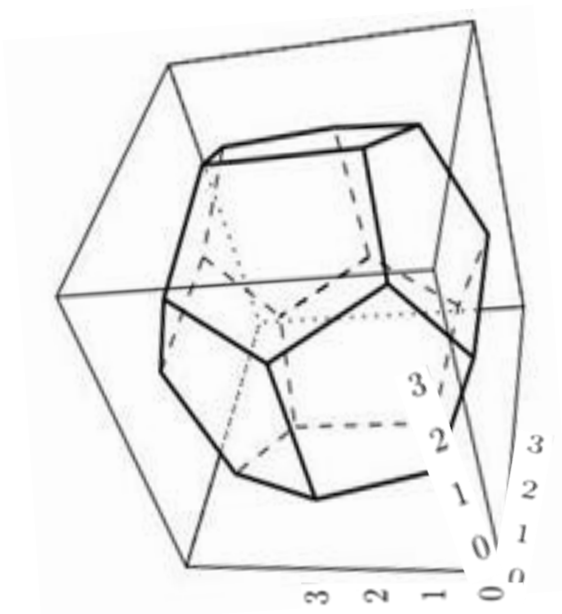
(Image from en.wikipedia.org/wiki/Steinitz_theorem )

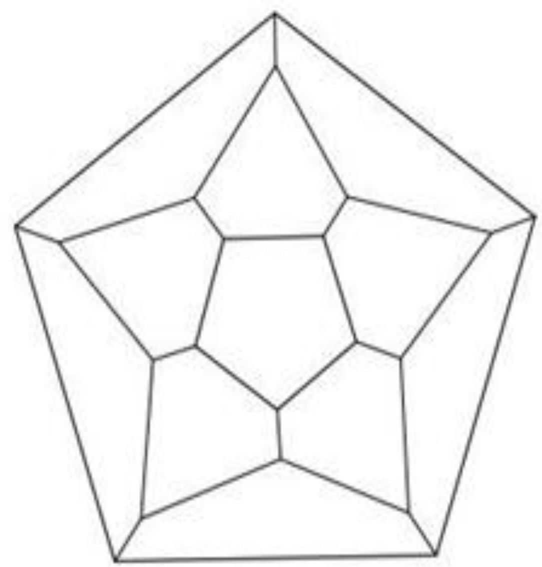
# Combinatorially Equivalent Polytopes: Intuition



**2 polytopes that are combinatorially equivalent to the cube and to each other**

# The idea of integer grid embeddings: Dodecahedron



# A citation that links polytope embedding to spanning trees

“To construct a 3-polytope with a **given combinatorial structure**, we follow the approach described in […] :
we construct a planar equilibrium embedding for a specified self-stress and lift it to a polyhedral surface via the Maxwell-Cremona correspondence. The analysis of the **determinant** of the linear system of equations which is used to define the equilibrium embedding
leads directly to the **number of spanning trees of the graph, via the Matrix-Tree theorem.** “

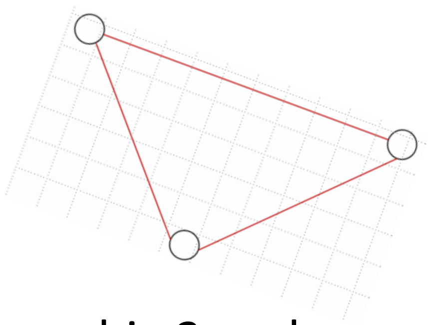
*from Rote, G.: “The number of spanning trees in a planar graph.” 2005.*

# Face cycles and triangles

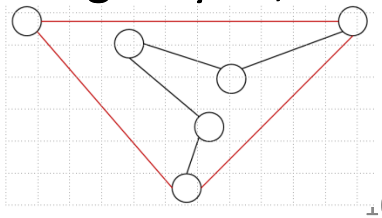
- A **face cycle** in a graph G is a cycle with edges in the boundary of a face of G.

- A face cycle can not surround “interior” vertices and edges.

- A face cycle  with three edges is called a **triangle**.



- Hence there exist 3-cycles, that are not  triangles ( e.g. a big 3-cycle, that contains other nodes and edges in its “interior”)



# Notations and Definitions

$\Gamma_n$ will denote the set of all planar graphs with n vertices.

For $G \in \Gamma_n$ let *t(G)* denote the number of its labeled spanning trees .

Let for all *n* *T(n)* be the maximal number of spanning trees that a G $\in \Gamma_n$ can have:
*T(n)* = $\max_{G \in \Gamma_n} \{t(G)\}$.

**We want to bound *T(n)* from above for growing n**; more precisely we search for an $\alpha$ such that for *n* large enough: *T(n)* $\leq \alpha^n$

# Triangulations

- If adding an edge destroys the planarity property, then the graph is called maximal planar.

- Such a graph has only faces, that are bounded by three edges, and it's called a triangulation.

- The graph that realizes the maximum *T(n)* must be a triangulation: adding edges to a planar graph leads to more spanning trees.

- Every plane triangulation with at least 4 vertices is also 3-connected.

- Hence only the (simple) planar triangulations with n vertices are examined.

# Planarity must be used

$5.\overline{3}^n$ was the best known upper bound.
It was proved by a method using the dual graph.

In the proof the triangulation property of the original graph is used, **but not its planarity property.**

**This leads to the main motivation of this paper:**
A proof method is used, that benefits both from the planarity and the triangulation of the graph family with n vertices.

This method is called **“refined outgoing edge approach”.**
**Why ‘refined’?:**
An older variant of the method failed to improve the old bound.

# The outgoing edge approach I

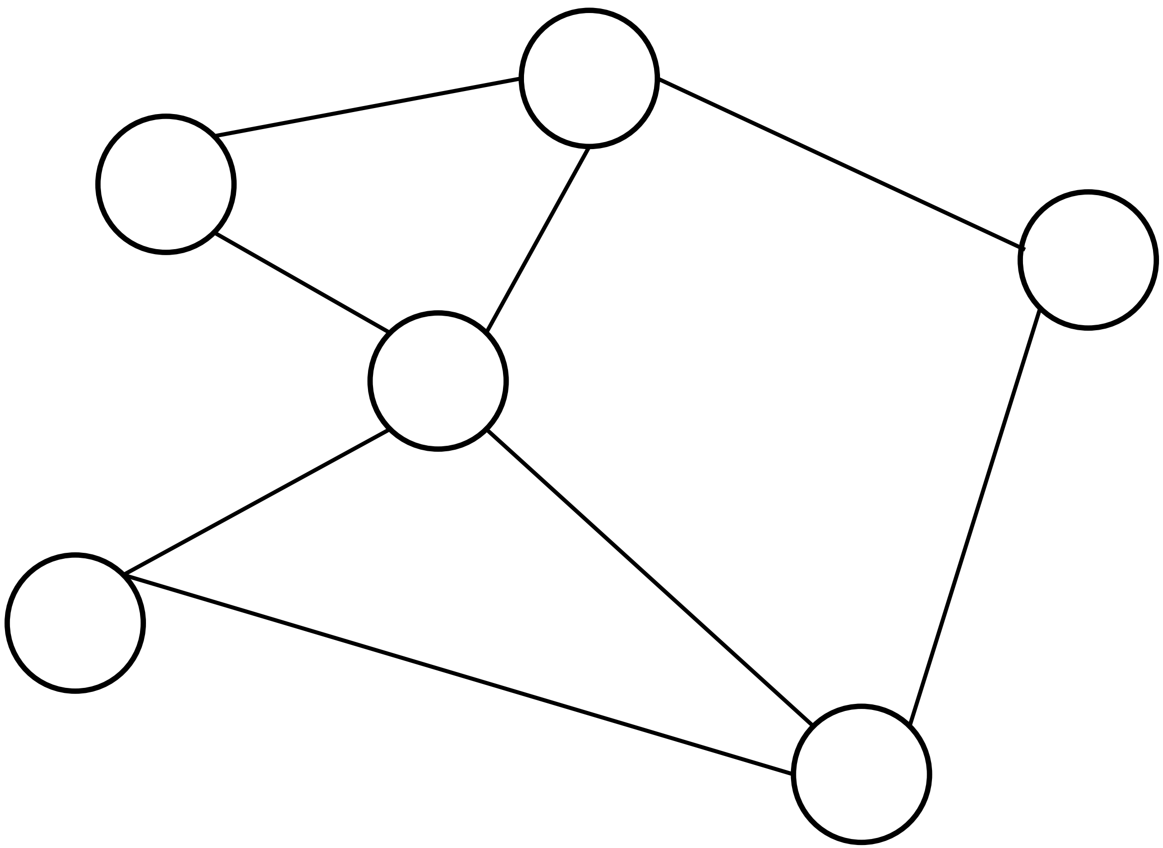
First we consider each edge of G **as a pair of directed arcs.**

**Definition:**
A directed graph is called **outdegree-one** if some designated vertex w has no outgoing arcs, and every other vertex is incident to exactly one outgoing arc. ❑

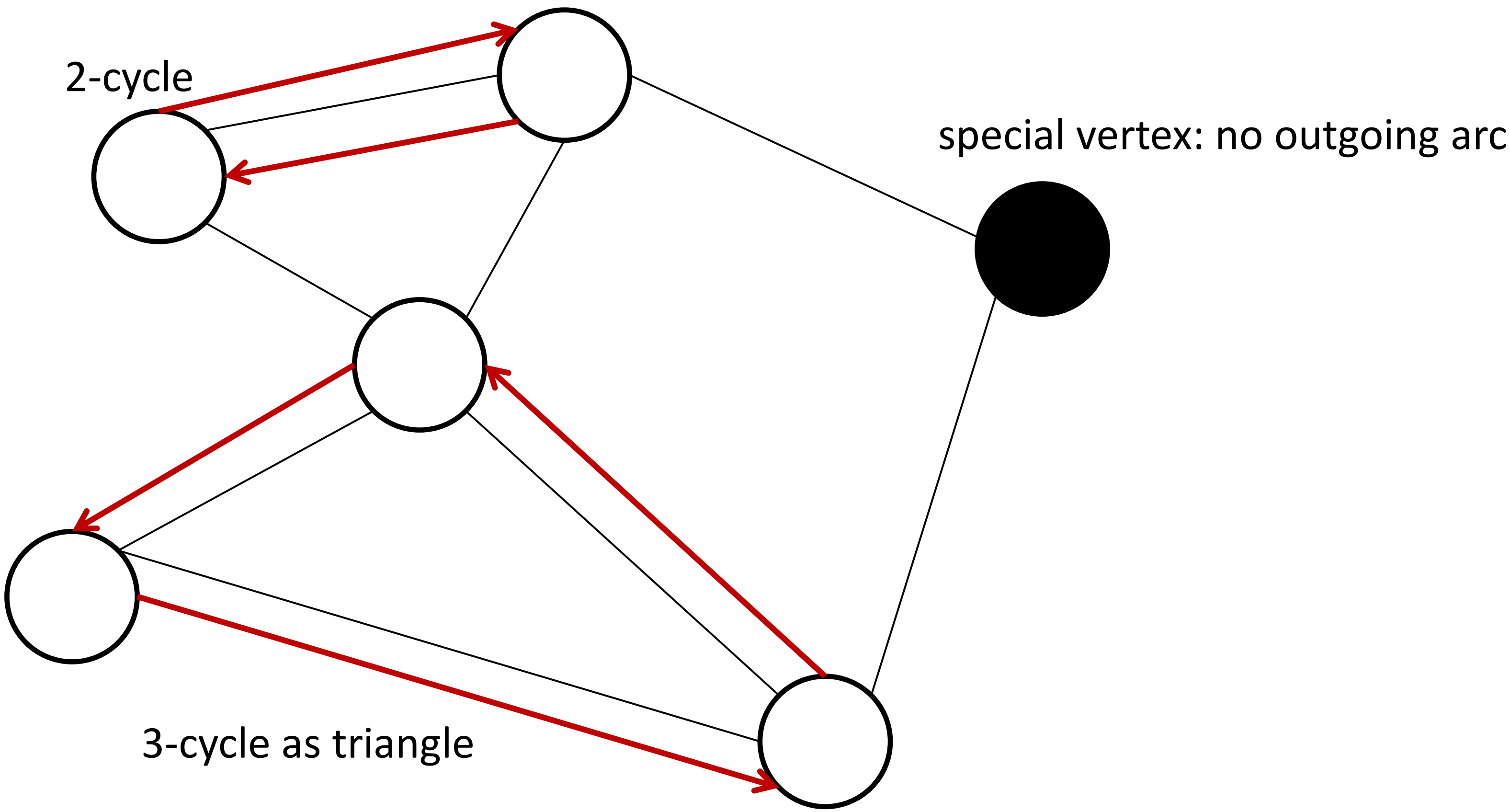
Every spanning tree of an undirected graph can -for a given special node w- be transformed into **exactly one associated outdegree-one graph** by selecting edges and orienting them as arcs towards w.

Since spanning trees contain all vertices, w can be fixed in advance.

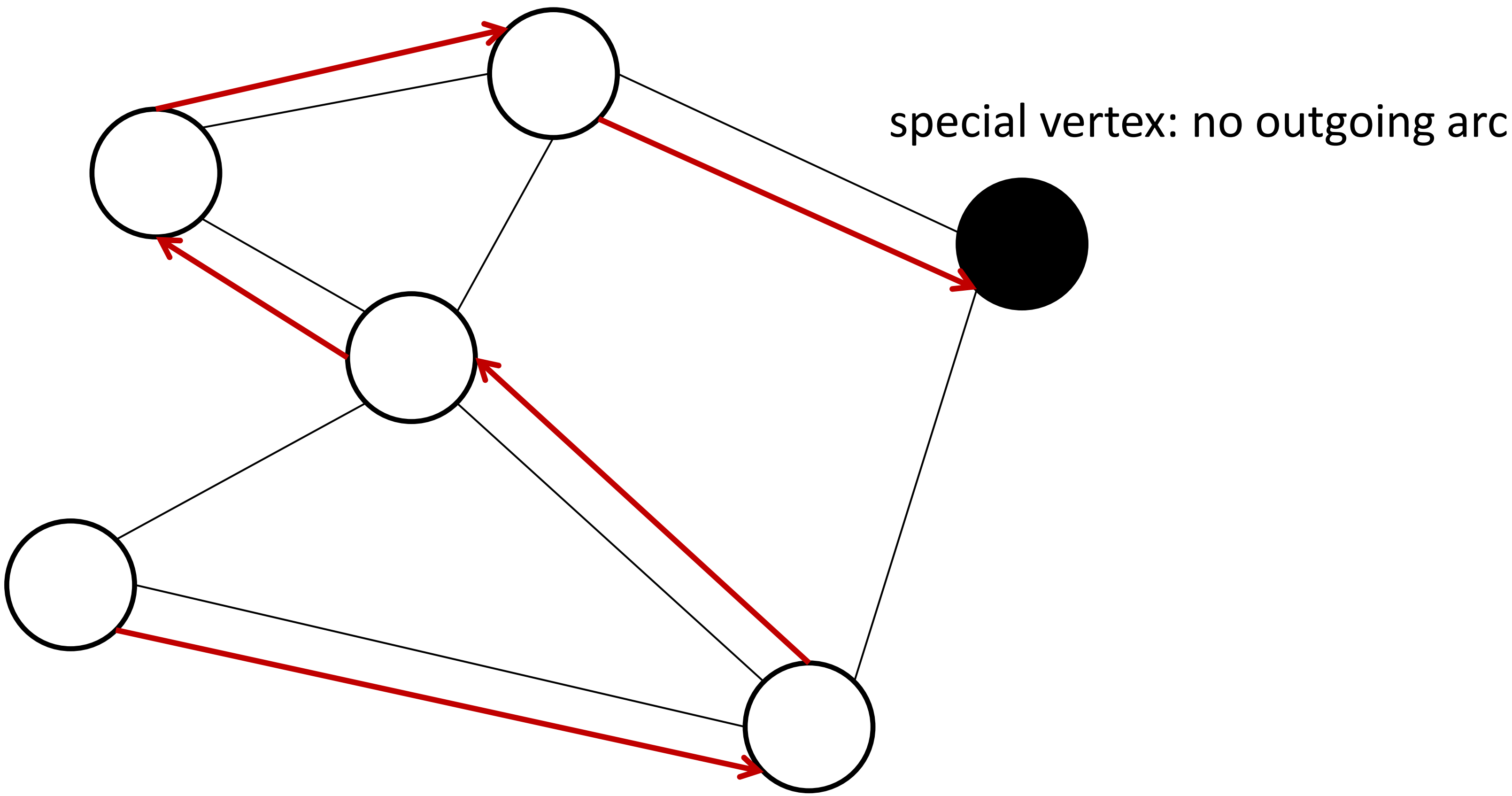
# Graph G (not a triangulation)



# Outdegree-one Graph w/ cycles



# Outdegree-one Graph w/o cycles: oriented spanning-tree



# The outgoing edge approach II

- Not every outdegree-one graph H is associated with a spanning tree, because H can contain directed cycles, if the original graph G contains associated undirected cycles.

- Especially edges of G could be transformed into a 2-cycle in H.

- Any cycle in H disconnects H.

# The outgoing edge approach III

**The number of different outdegree-one graphs (contains # of spanning trees):**
Let $d_i$ be the degree of the vertex $v_i$ in G. For every $v_i$ we have $d_i$ choices how to select ist outgoing edge.

So we have $\prod_{i=2}^{n} d_i$ different outdegree-one graphs in G.

This is a number smaller **than $6^n$** (6 is coming from a simpler approach), **if G is planar.**

**But :** the simple approach gives no improvement over $5.\overline{3}^n$.

# The outgoing edge approach IV

Now a basic idea:
A spanning tree is a maximal set of edges, that contains no cycle.

**Outdegree-one graphs without cycles are exactly the oriented spanning trees of G.**

Let $P_{nc}$ be the probability that the random outdegree-one graph generated by picking an outgoing edge for every vertex of G uniformly at random contains no
cycle.
Then: $t(G) = (\prod_{i=2}^{n} d_i) P_{nc}$ where G is not necessary planar.

# Cycles and Outdegree-One Graphs I

- Assume that G contains s cycles, enumerated in some order.
- Let $\mathbf{C_i}$ be the event that the i-th cycle occurs in a random outdegree-one graph, let $\mathbf{C^c_i}$ be the event that the i-th cycle does not occur.

- Two cycles are independent **:** ⇔ They do not share a vertex.
- The two events $\mathbf{C_i}$ and $\mathbf{C_j}$ are independent (dependent)
  ⇒ The two corresponding cycles are independent (dependent).

**Definition :**
The events $C_1, \ldots, C_l$ have **mutually exclusive dependencies**,
if the dependency of $(C_i, C_j)$ implies $\Pr[C_i \cap C_j] = 0$.
❑

**Definition :**
The events $C_1, \ldots, C_l$ have **union-closed independencies**,
if the independence of the pairs $(C_i, C_{j1}), \ldots, (C_i, C_{jk})$ implies
independency of $C_i$ and $(C_{j1} \cup \ldots \cup C_{jk})$ ❑

# Cycles and Outdegree-One Graphs II

All events $C_i$ obviously have mutually exclusive dependencies and union-closed independencies.

This leads to the following

**Lemma (Main Lemma):**
The events $C_1, \ldots, C_l$ have mutually exclusive dependencies and union-closed
independencies, and from this follows for $1 < k < l$

$$\Pr[\bigcap_{j=k}^{l} C_j^c \mid \bigcap_{i=1}^{k-1} C_i^c] \leq \prod_{j=k}^{l} \left(1 - \frac{\Pr[C_j]}{\prod_{\substack{1 \leq i < k: \\ C_i, C_j \\ dependent}} \Pr[C_i^c] \sqrt{\prod_{\substack{k \leq i \leq l: \\ C_i, C_j \\ dependent}} \Pr[C_i^c]}}\right)$$

❑

# Probability of cycles and vertex degrees

- The idea behind the Main Lemma is to express $P_{nc}$ in such away that it is equal to the left hand side of the inequation.

- The rhs then gives an upper bound on $P_{nc}$ .

- The rhs of the inequation is easier to handle if we know the probabilities
  $Pr[C_i]$ or $(1 - Pr[C_i])$.

- But combinatorially it holds that
  $Pr[C_i] = 1 / (d_a d_b)$ if the i-th cycle is a 2-cycle on the vertices $v_a, v_b$.

- $Pr[C_j] = 2 / \prod_{a: v_a \in Z} d_a$ if the i-th cycle is at least a 3-cycle on the set Z.

# Bounding the probability of cycles appearing I

- Let s be the number of cycles in G after introducing 2 arcs per edge. Then

$$P_{nc} = \Pr[\bigcap_{j=1}^{s} C_j^c]$$

- The rhs of the Main Lemma also needs information about the dependency of cycles.

- Therefore the computation is restricted to 2-cycles and 3-cycles **that are triangles.**

- Discarding larger cycles is possible:
  including more cycle lengths lead to a smaller $P_{nc}$ , and so
  we still have an upper bound after discarding.

- Let $s_2$ be the number of 2-cycles of G, and $s_3$ the number of triangles.
  Then we consider $s := s_2 + s_3$ cycles.

# Bounding the probability of cycles appearing II

- Matching the left hand side of the Main Lemma:

$$\Pr[\bigcap_{j=s_3+1}^{s} C_j^c \mid \bigcap_{j=1}^{s_3} C_j^c]$$

**This is the probability that no 2-cycle occurs under the assumption that no triangle occurred.**

- Another matching of the left hand side is ($k = 1$, $l = s_3$):

$\Pr[\bigcap_{j=1}^{s_3} C_j^c]$ **, the probability that no triangle occurs .**

However, **the logarithmic version** of $t(G) = (\prod_{i=2}^{n} d_i) P_{nc}$ will be bounded:

$$\log t(G) = \sum_{i=2}^{n} \log d_i + \log \Pr[\bigcap_{j=1}^{s} C_j^c].$$

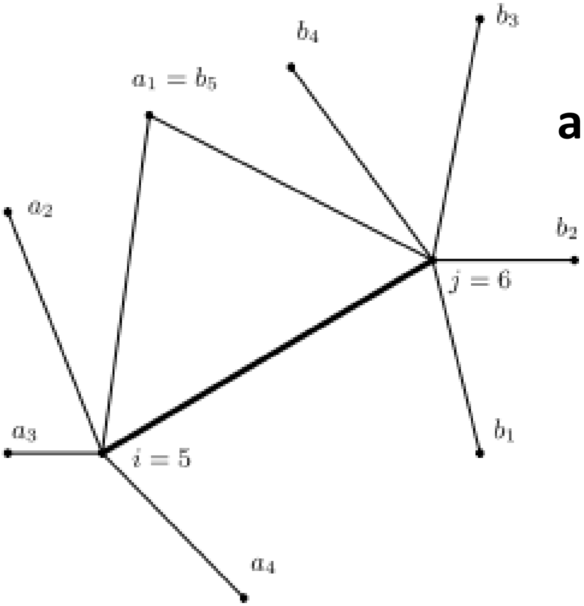
# Bounding the probability of cycles appearing III

In total we then have from the Main Lemma:

$$\mathrm{P}_{\mathrm{nc}} \leq \Pr[\bigcap_{j=1}^{s} C_j^c]$$

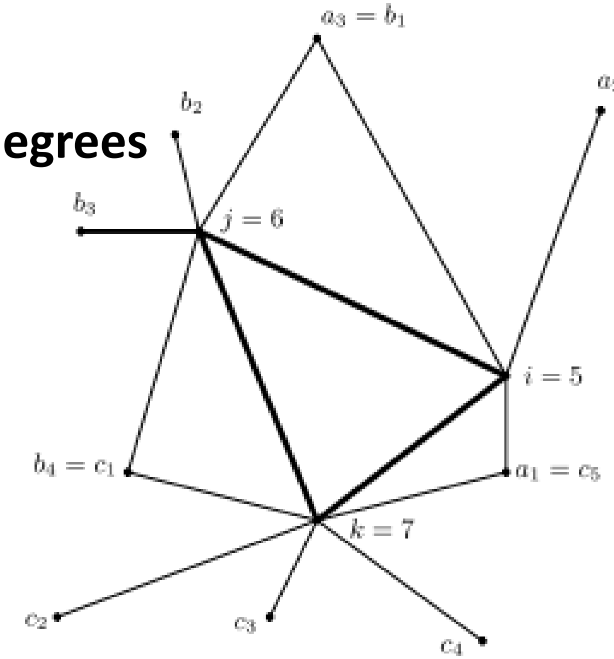
$$\leq \sum_{j=1}^{s_3} \log\left(1 - \frac{\Pr[C_j]}{\sqrt{\prod_{\substack{1 \leq i \leq s_3: \\ C_i, C_j \\ dependent}} \Pr[C_i^c]}}\right) + \sum_{j=s_3+1}^{s} \log\left(1 - \frac{\Pr[C_j]}{\prod_{\substack{1 \leq i < s_3+1: \\ C_i, C_j \\ dependent}} \Pr[C_i^c] \sqrt{\prod_{\substack{s_3 < i \leq s: \\ C_i, C_j \\ dependent}} \Pr[C_i^c]}}\right)$$

# Planarity occurs for the first time: 2-extensions of 2- and 3-cycles




$(5, 6, (a_1, \ldots, a_4), (b_1, \ldots, b_5))$

**$a_i$, $b_i$, $c_i$ are vertex degrees**




$(5, 6, 7, (a_1, \ldots, a_3), (b_1, \ldots, b_4), (c_1, \ldots, c_5))$

Convention for naming the signatures of 2-extensions of 2-cycles (on the left) and 3-cycles (on the right).

In the tuples we have a **uniquely determined clockwise ordering** of degrees.
Possible because G is **planar and 3-connected**:
**If a graph is planar and 3-connected its facial structure is uniquely determined (Whitney).**

# Transforming the problem into a linear program I

**The main idea:**
construct a linear programming problem using the signatures of the 2-extensions of the 2-cycles/triangles

**Why does this make sense?**
Because the signatures of the 2-extensions of the cycles can be used to describe whether two cycles are dependent.

To achieve this, the sum above is rewritten, such that cycles with identical signatures are grouped together.

# Transforming the problem into a linear program II

$$P_{nc} \leq \Pr[\bigcap_{j=1}^{s} C_j^c]$$

$$\leq \sum_{j=1}^{s_3} \log\left(1 - \frac{\Pr[C_j]}{\sqrt{\prod_{\substack{1 \leq i \leq s_3: \\ C_i, C_j \\ \underline{dependent}}} \Pr[C_i^c]}}\right) + \sum_{j=s_3+1}^{s} \log\left(1 - \frac{\Pr[C_j]}{\prod_{\substack{1 \leq i < s_3+1: \\ C_i, C_j \\ \underline{dependent}}} \Pr[C_i^c] \sqrt{\prod_{\substack{s_3 < i \leq s: \\ C_i, C_j \\ \underline{dependent}}} \Pr[C_i^c]}}\right)$$

**translated into** $\sum_{i,j,k,A,B,C} f_{ijk}(A,B,C) \log P_{ijk}(A,B,C) + \sum_{i,j,A,B} f_{ij}(A,B) \log P_{ij}(A,B)$

1. The new sums range over **all feasible signatures**
2. The **number** of 2-extensions of 3-cycles with signature (i,j,k,A,B,C) is denoted with $f_{ijk}(A,B,C)$
3. The **number** of 2-extensions of 2-cycles with signature (i,j,A,B) is denoted with $f_{ij}(A,B)$
4. **These f-variables will be the variables of our LP-Program**

# Coefficients of the objective function = Dependency of cycles expressed with signatures

$$P_{ij}(A,B) :=$$

$$1-\frac{1}{ij\prod_{1\le p\le i-2}\left(1-\frac{2}{ia_p a_{p+1}}\right)\prod_{1\le p\le j-2}\left(1-\frac{2}{jb_p b_{p+1}}\right)\left(1-\frac{2}{ija_1}\right)\left(1-\frac{2}{ijb_1}\right)\sqrt{\prod_{1\le p\le i-1}\left(1-\frac{1}{ia_p}\right)\prod_{1\le p\le j-1}\left(1-\frac{1}{jb_p}\right)}}$$

$$P_{ijk}(A,B,C) :=$$

$$1-\frac{2}{ijk\sqrt{\prod_{1\le p\le i-2}\left(1-\frac{2}{ia_p a_{p+1}}\right)\prod_{1\le p\le j-2}\left(1-\frac{2}{jb_p b_{p+1}}\right)\prod_{1\le p\le k-2}\left(1-\frac{2}{jc_p c_{p+1}}\right)\left(1-\frac{2}{ika_1}\right)\left(1-\frac{2}{ijb_1}\right)\left(1-\frac{2}{jkc_1}\right)}}$$

**These are probabilities that certain 2-cycles and 3-cycles do not occur**

**$P_{ij}$ / $P_{ijk}$ are no variables, since they belong to $f_{ij}$ / $f_{ijk}$**
**and hence they are fixed by i,j,k,A,B,C, and are *coefficients***

$$\log t(G) = \sum_{i=2}^{n} \log d_i + \log \Pr[\bigcap_{j=1}^{s} C_j^c]$$ must be bounded from above.

Now we also want to rewrite $\sum_{i=2}^{n} \log d_i$ using the f-variables.

For convenience we use $\sum_{i=2}^{n} \log d_i + d_1$.

**Idea:** if we can express the whole sum log t(G) with the f-variables, then an objective function for the LP can be obtained.

Let $D := \sum_{i=1}^{n} \log d_i .$

Now the f-variables, which are signature counters, must come into play.

⇒ we distribute the whole sum D to different types of signatures.

Let $0 \leq \mu_i \leq 1$, i = 1, …,4. Let $\sum_{i=1}^{4} \mu_i = 1.$

Then D can be split into four parts (as a weighted sum):
$D_i := \mu_i D$, i = 1, … , 4

- Every vertex $v_a$ is part of $d_a$ 2-cycles.

- Every vertex $v_a$ is part of $d_a$ triangles (triangulation)

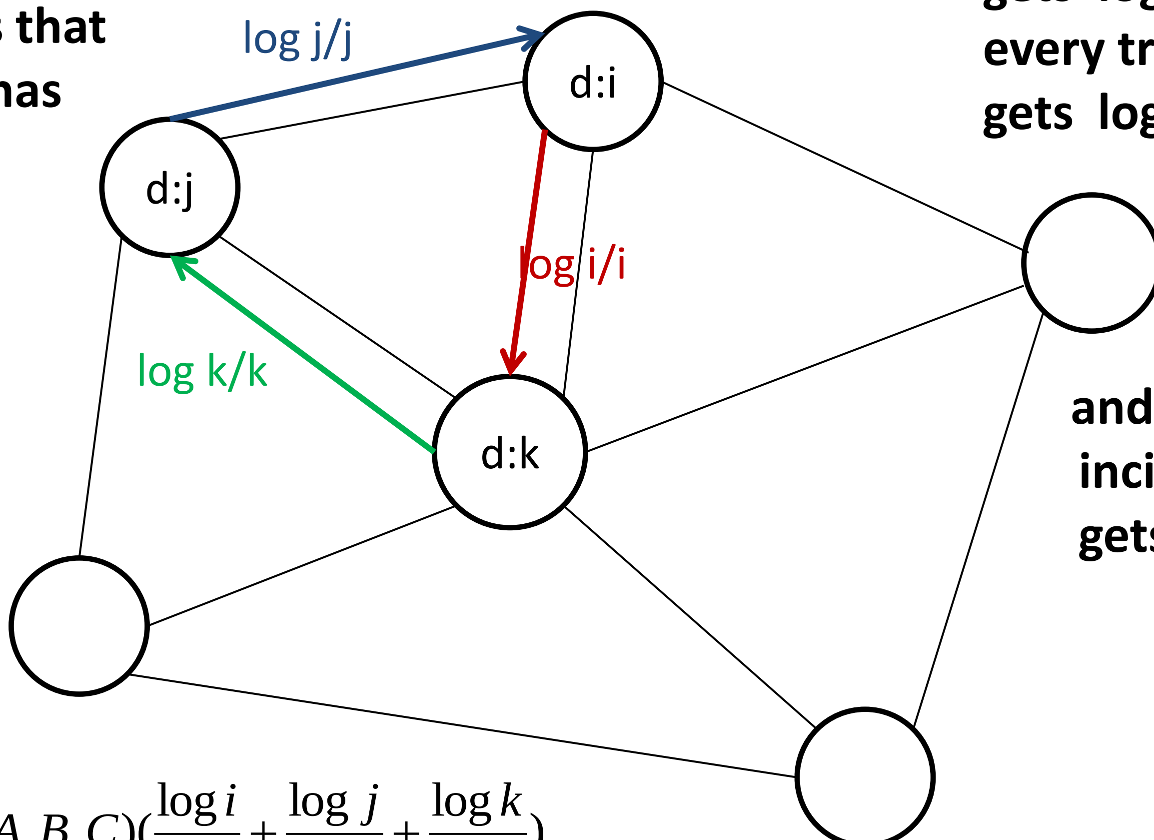
- Every vertex is part of a 2-extension of a 2- or 3-cycle(triangle)

⇒The log-sum of the degrees D can be reconstructed in 4 ways by using 2-cycles, 2-extensions of 2-cycles, triangles and 2-extensions of triangles.

This is what is called a **charging scheme**:

1. the log $d_i$ are distributed via the **edges** of a vertex (**edges are charged**)
2. what every 2-cyle, 3-cyle or 2-extension of a cycle gets from his incident neighbors sums up to its “personal” portion of the log-sum D.
3. the portions are grouped into identical signatures

# Charging scheme example: triangles




**d:i means that vertex $v_i$ has degree i.**

**Every triangle incident to $v_i$ gets log i/i from $v_i$ , every triangle incident to $v_j$ gets log j/j from $v_j$**

**and every triangle incident to $v_k$ gets log k/k from $v_k$ .**

$$\sum_{i,j,k,A,B,C} f_{ijk}(A,B,C)\left(\frac{\log i}{i}+\frac{\log j}{j}+\frac{\log k}{k}\right)$$

# Charging scheme for 2-extensions of 2-cycles

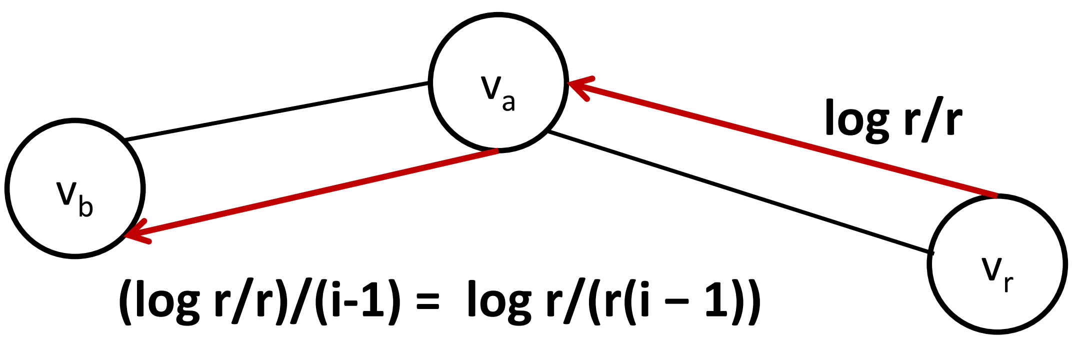
**Example: 2-extensions of 2-cycles**

Let $v_a$ $v_b$ be an edge in G and let $v_r \neq v_b$ be a vertex adjacent to $v_a$ .

Let $d_a$= i. Let $d_r$= r.

Distributing log $d_r$ uniformly, assigns every 2-extension with "endpoint" $v_r$ the fraction of

**log $d_r$ /($d_r$ (i – 1)) = log r/(r(i – 1))** from $v_r$ .




$$\sum_{i,j,A,B} f_{ij}(A,B)\left(\sum_{a_r \in A} \frac{\log a_r}{a_r(i-1)} + \sum_{b_r \in B} \frac{\log b_r}{b_r(j-1)}\right)$$

$D_i := \mu_i D,\ i = 1, \dots, 4$ or alternatively $D_i := \mu_i \sum_{i=1}^{n} \log d_i .$

$D_1$ has a relation to 2-cycles, $D_2$ has a relation to 2-extensions of 2-cycles, $D_3$ has a relation to triangles, $D_4$ has a relation to 2-extensions of triangles.

The $\mu_i$ –parameters are **estimators** of the fraction that 2-cycles contribute to the whole log-sum, of the fraction that triangles contribute to the whole log-sum and so on.

These parameters will be fixed later, based on experimental data.
$\mu$-values used by the authors were (for i = 1, … ,4): 0.3, 0.25, 0.225, 0.225.

From this we get:

$D_1 = \mu_1 \cdot$ ( D = $\sum_{i=1}^{n} \log d_i$ only expressed with the help of 2-cycles)

$$= \mu_1 \cdot \sum_{i,j,A,B} f_{ij}(A,B)\left(\frac{\log i}{i} + \frac{\log j}{j}\right)$$

$D_2 = \mu_2 \cdot$ ( D = $\sum_{i=1}^{n} \log d_i$ only expressed with the help of 2-extensions of 2-cycles)

$$= \mu_2 \cdot \sum_{i,j,A,B} f_{ij}(A,B)\left(\sum_{a_r \in A} \frac{\log a_r}{a_r(i-1)} + \sum_{b_r \in B} \frac{\log b_r}{b_r(j-1)}\right)$$

$D_3 = \mu_3 \cdot$ ( D = $\sum_{i=1}^{n} \log d_i$ only expressed with the help of 3-cycles)

$$= \mu_3 \cdot \sum_{i,j,k,A,B,C} f_{ijk}(A,B,C)\left(\frac{\log i}{i} + \frac{\log j}{j} + \frac{\log k}{k}\right)$$

$D_4 = \mu_4 \cdot$ ( D = $\sum_{i=1}^{n} \log d_i$ only expressed with the help of 2-extensions of 3-cycles)

$$= \mu_4 \cdot \sum_{i,j,k,A,B,C} f_{ijk}(A,B,C)\left(\sum_{a_r \in A} \frac{\log a_r}{a_r(i-1)} + \sum_{b_r \in B} \frac{\log b_r}{b_r(j-1)} + \sum_{c_r \in C} \frac{\log c_r}{c_r(k-1)}\right)$$

# Objective Function

$\log t(G) = \sum_{i=2}^{n} \log d_i + \log \Pr[\bigcap_{j=1}^{s} C_j^c]$ can now be bounded from above

using signatures and f-counting-variables, and the upper bound is the objective function of an LP and thus has to be maximized.

The objective function is split into an $f_{ij}$-part and an $f_{ijk}$-part:

(G2) $$\max D_1 + D_2 + \sum_{i,j,A,B} f_{ij}(A,B) \log P_{ij}(A,B)$$

(G3) $$\max D_3 + D_4 \sum_{i,j,k,A,B,C} f_{ijk}(A,B,C) \log P_{ijk}(A,B,C)$$

Both optimal values contribute to the complete solution.

# Constructing LP-constraints: planarity is used for the 2nd time

**Necessary** LP-constraints that hold for n-vertex planar graphs are:

$$\sum_{i,j,A,B} f_{ij}(A,B)\left(\frac{1}{i}+\frac{1}{j}\right)=n \qquad \textbf{(1)}$$

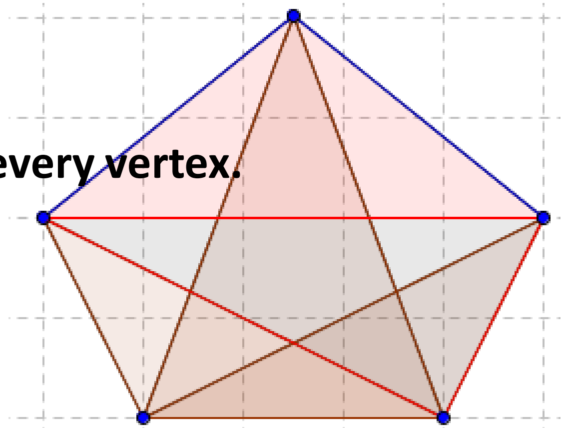
$$\sum_{i,j,A,B} f_{ij}(A,B)\left(\sum_{a_r\in A}\frac{1}{a_r(i-1)}+\sum_{b_r\in B}\frac{1}{b_r(j-1)}\right)=n \qquad \textbf{(2)}$$

$$\sum_{i,j,k,A,B,C} f_{ijk}(A,B,C)\left(\frac{1}{i}+\frac{1}{j}+\frac{1}{k}\right)=n \qquad \textbf{(3)}$$

$$\sum_{i,j,k,A,B,C} f_{ijk}(A,B,C)\left(\sum_{a_r\in A}\frac{1}{a_r(i-1)}+\sum_{b_r\in B}\frac{1}{b_r(j-1)}+\sum_{c_r\in C}\frac{1}{c_r(k-1)}\right)=n \qquad \textbf{(4)}$$

**Counter-Example**

**$K_5$ : 10 “triangles” charged with 0.75 each total sum = 7.5 > n**



**This is a re-usage of the charging scheme with charge 1 for every vertex.**

**Why must this hold for planar graphs?**
**Edges are charged. <u>What exactly do they delimit?</u>**

# Necessary Constraints II

Every 2-cycle is counted by some $f_{ij}$-variable; let m be the number of edges:

$$\sum_{i,j,A,B} f_{ij}(A,B) \leq m \qquad \textbf{(5)}$$

Only 3-cycles that are triangles are considered, hence the sum over all $f_{ijk}$-variables equals the number of triangles, which in planar graphs is at most 2n.

$$\sum_{i,j,k,A,B,C} f_{ijk}(A,B,C) \leq 2n \qquad \textbf{(6)}$$

From Euler's formula:

$$m \leq 3n \qquad \textbf{(7)}$$

# Necessary Constraints III

- Necessary constraints:
  G planar $\Rightarrow$ Constraint must hold or
  Constraint does not hold $\Rightarrow$ G is not planar.

- So by constraints (1) – (7) we keep all planar graphs in the set of graphs, that fulfill the constraints.
  (1) – (7) hold **at least** for all planar graphs.

- There might be graphs that are not planar, but fulfill (1) – (7).
  Only interesting, if they disturb the upper bound by making it greater.

- **Some** of these non-planar graphs can be excluded step by step using further constraints.
  But only in one special case this gave an improvement of the upper bound.

# 2 LP's

**LP-2C derived from the 2-cycles:**

Maximize $D_1 + D_2 + \sum_{i,j,A,B} f_{ij}(A,B)\log P_{ij}(A,B)$ under the constraints

(1), (2), (5) and (7)

**LP-3C derived from the 3-cycles:**

Maximize $D_3 + D_4 + \sum_{i,j,k,A,B,C} f_{ijk}(A,B,C)\log P_{ijk}(A,B,C)$ under the constraints

(3),(4),(6)

# Roadmap for linear optimization

So far we can maximize the number of spanning trees of a planar graph with **n vertices** t(G) by solving the two LP's.

But:
we want to have a solution that is asymptotical for growing n
⇒ we normalize the LP's : **divide them by n**
⇒ n is not contained in the problem anymore

**Consequences:**
all information contained in n is lost
**-** dividing by n eliminates the largest possible vertex degree of n-1
- dividing by n leads to infinitely many possible signatures and thus to an LP with infinitely many f-variables

**Solution:**
Solve the dual LP.

# Roadmap part II: Duality

The dual problem is a minimization, where we have 3 variables $\lambda_i$, i= 1,…,3:
one for each primal constraint,
but now there are infinitely many dual constraints.

Weak duality theorem:
any point in the feasible area of the dual problem is an upper bound for the primal solution
⇒ find such a feasible point !

# Roadmap part III: Solving the Dual Problem

How to solve the dual program with infinitely many constraints?

1. Make some 'good' choices for the $\mu_i$ values.

2. $f_{ij}$ with i,j far away from 6 will probably be zero
   $\Rightarrow$ the corresponding constraints in the dual problem will not reach equality.

3. Under assumptions 1., 2. construct a problem with only **finitely many constraints** and find a feasible point for these constraints.

4. Now there is a 'good' chance, that all other constraints also hold for this feasible point.

5. **Very tedious job:**
   **PROVE that ALL dual constraints are fulfilled for the candidate point.**

e^(0.180948 + (0.232445 * 3) + 0.0980332 + 0.192612 + (0.247984 * 2)) = **5.2851**

**This computation yields $\alpha$ because $\log \alpha^n = n \log \alpha$, and we had divided by n.**

| LP instance | $\mu$ values | solution | $\alpha$ |
|---|---|---|---|
| general problem (2-cycles) (G2)→ max, s.t. (A2-C2), $m = 3$ | $\mu_1 = 0.3, \mu_2 = 0.25$ | $\lambda_2 = 0.180948$ $\lambda_3 = 0.232445$ | 5.28515 |
| general problem (3-cycles) (G3)→ max, s.t. (A3-C3) | $\mu_3 = 0.225, \mu_4 = 0.225$ | $\lambda_1 = 0.0980332$ $\lambda_2 = 0.192612$ $\lambda_3 = 0.247984$ | |
| restricted problem (no triangles) (R2)→ max, s.t. (A2-C2), $m = 2$ | $\mu_1 = 0.9, \mu_2 = 0.1$ | $\lambda_3 = 0.614264$ | 3.41619 |
| restricted problem (no 3,4-gons) (R2)→ max, s.t. (A2-C2,E33) $m = 5/3$ | $\mu_1 = 0.615, \mu_2 = 0.385$ | $\lambda_1 = -0.615054$ $\lambda_3 = 0.744706$ $\lambda_4 = 0.001954$ | 2.71567 |

$5.\overline{3}^n$

Table 1: The results of the dual linear programs. Not specified $\lambda$ values are zero.

A 3-connected planar graph must contain always a triangle, a quadrilateral face or a pentagonal face $\longrightarrow$ leads to restricted problems

# Main Results

**Main Theorem:**

Let G be a planar graph with n vertices. The number of spanning trees of G is at most $O(5.28515^n)$.
If G is 3-connected and contains no triangle, then the number of its spanning trees is bounded by $O(3.41619^n)$.
If G is 3-connected and contains no triangle and no quadrilateral, then the number of its spanning trees is bounded by $O(2.71567^n)$. ❑

**Corollary (3D polytope integer grid embedding):**
The grid size needed to realize a 3D polytope with integer coordinates is bounded by $O(147.7^n)$ (best former bound $O(188^n)$).
For grid embeddings of simplicial 3d polytopes the bound is $O(27.94^n)$ (best former bound $O(28.444\ldots^n)$).
❑

# Limitations of the Outgoing-edge approach

It is not easy to get better bounds with this approach:

1. Additionally exclude larger cycles? (yields a smaller and more exact $P_{nc}$)
   - leads to a much more complicated formulation of the LP problem (case distinctions)
   - verifying all dual constraints for candidate points is probably intractable

2. The outgoing-edge approach is local. Make it more global by analysing ‘extensions of extensions’?
   - same drawback as above: probably too complicated

3. Using other enumeration schemes for enumerating the cycles? Might lead to an improvement.

4. Using further constraints in the primal LP’s?

# Summary I

- The specific idea of the refined outgoing-edge approach lies in the introduction of **signatures for triangulations .**

- Since we only deal with planar triangulations it is possible to define such signatures which describe the local environment of every node:

  How - in terms of node degrees of the neighbors - do the 2-cycles, triangles, 2-extensions of 2-cycles and 2-extensions of triangles for every vertex look like?

- **This gives us an abstract description of the local environment of every vertex, without looking at any concrete graph.**

- Now it is assumed that an outdegree-one graph is randomly chosen. **Is it a spanning tree or not?**

# Summary II

- By grouping the identical signatures into counting variables, the desired property, **that a random selection of outgoing edges must not contain cycles,** can be expressed **in the signature counters**.

- The basic inequality that gives an upper bound on the number of spanning trees comes from the rhs of the Main Lemma, which then is 1:1 translated into
  **a term of signature counting variables and their coefficients.**

- The coefficients are numbers that can be computed from the signature parameters, and they contain the 'exclude-the-cycles' condition.

- Hence the rhs of the Main Lemma is translated into a maximizing Linear Program, with the signature counters as variables.

- The condition that only planar graphs are examined is packed into the LP-constraints.

# Summary III

- The whole approach is very generous, because almost all cycles are **not** excluded, and also some non-planar graphs are not excluded by the LP-constraints.

- Nevertheless the old bound is improved from $(5+(1/3))^n$ to $5.28515^n$.

- **How can one find an upper bound on the number of spanning trees in planar graphs, or more precisely in triangulations?**

  **By looking at and counting small local neighborhoods of vertices in outdegree-one graphs with identical signatures , ensuring that at least small cycles are excluded,**
  **<u>hoping</u> that the bound, i.e. the objective value of the LP will be better than $(5+(1/3))^n$ , which luckily was the case.**

# References

[1] Buchin, K; Schulz, A.: “On the Number of Spanning Trees a Planar Graph Can Have“, Algorithms - 18th Annual European Symposium (ESA 2010), Lecture Notes in Computer Science, 6346, Springer-Verlag, pp. 110–121, 2010
(**full version** at arXiv.org:
arXiv:0912.0712v2 [math.CO], 5 Sep 2010)

[2] Ribó Mor, A.: “Realization and Counting Problems for Planar Structures: Trees and Linkages, Polytopes and Polyominoes.” PhD thesis, Freie Universität Berlin, 2006

[3] Rote, G.: “The number of spanning trees in a planar graph.” In Oberwolfach Reports, European Mathematical Society, Publishing House, 2005.

[4] Ribó Mor, A.;Rote,G.;Schulz,A.: Small grid embeddings of 3-polytopes, arXiv:0908.0488v2 [cs.CG] , 23 Jul 2010

# Transforming the problem into a linear program III

For better readability we introduce the following notations ($X$ is used as a placeholder for $A, B$, or $C$, and $x$ as a placeholder for $a, b$, or $c$):

$$
\begin{aligned}
P_2(r, X) &:= \prod_{1 \le p \le r-1} \left(1 - \frac{1}{r x_p}\right), \quad P_3(r, X) := \prod_{1 \le p \le r-2} \left(1 - \frac{2}{r x_p x_{p+1}}\right), \\
P_{ij}(A, B) &:= 1 - \frac{1}{ij P_3(i, A) P_3(j, B) \left(1 - \frac{2}{ija_1}\right)\left(1 - \frac{2}{ijb_1}\right)\sqrt{P_2(i, A) P_2(j, B)}}, \\
P_{ijk}(A, B, C) &:= 1 - \frac{2}{ijk\sqrt{P_3(i, A) P_3(j, B) P_3(k, C)\left(1 - \frac{2}{ika_1}\right)\left(1 - \frac{2}{ijb_1}\right)\left(1 - \frac{2}{jkc_1}\right)}}.
\end{aligned}
$$

# Dual Problems written down

**General Problem (2-cycle part):**

$$\text{Minimize } \lambda_1 + \lambda_2 + 3\lambda_3,$$

such that, $\lambda_3 \geq 0$, and for all signatures $(i, j, A, B)$:

$$\log P_{ij}(A,B) + \mu_1\Big(\frac{\log i}{i} + \frac{\log j}{j}\Big) + \mu_2\Big(\sum_{a_r \in A} \frac{\log a_r}{a_r(i-1)} + \sum_{b_r \in B} \frac{\log b_r}{b_r(j-1)}\Big)$$
$$- \lambda_1\Big(\frac{1}{i} + \frac{1}{j}\Big) - \lambda_2\Big(\sum_{a_r \in A} \frac{1}{a_r(i-1)} + \sum_{b_r \in B} \frac{1}{b_r(j-1)}\Big) - \lambda_3 \leq 0. \quad (10)$$

**General Problem (3-cycle part):**

$$\text{Minimize } \lambda_1 + \lambda_2 + 2\lambda_3,$$

such that, $\lambda_3 \geq 0$, and for all signatures $(i, j, k, A, B, C)$:

$$\log P_{ijk}(A,B,C) + \mu_3\Big(\frac{\log i}{i} + \frac{\log j}{j} + \frac{\log k}{k}\Big) + \mu_4\Big(\sum_{a_r \in A} \frac{\log a_r}{a_r(i-1)} + \sum_{b_r \in B} \frac{\log b_r}{b_r(j-1)} + \sum_{c_r \in C} \frac{\log c_r}{c_r(k-1)}\Big)$$
$$- \lambda_1\Big(\frac{1}{i} + \frac{1}{j} + \frac{1}{k}\Big) - \lambda_2\Big(\sum_{a_r \in A} \frac{1}{a_r(i-1)} + \sum_{b_r \in B} \frac{1}{b_r(j-1)} + \sum_{c_r \in C} \frac{1}{c_r(k-1)}\Big) - \lambda_3 \leq 0$$