The number of labeled graphs of bounded treewidth

Julien Baste[1] Marc Noy[2] Ignasi Sau[1]

Seminário ParGO, UFC
Fortaleza, September 16, 2016

[1] CNRS, LIRMM, Université de Montpellier, Montpellier, France, and Departamento de Matemática, UFC, Fortaleza, Brazil

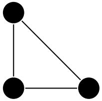
[2] Department of Mathematics of Universitat Politècnica de Catalunya, and Barcelona Graduate School in Mathematics, Barcelona, Catalonia

[arXiv 1604.07273]

Next section is...

k-trees and partial k-trees

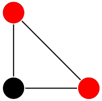
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

k-trees and partial k-trees

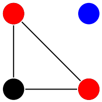
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

k-trees and partial k-trees

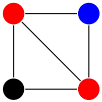
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

k-trees and partial k-trees

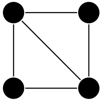
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

k-trees and partial k-trees

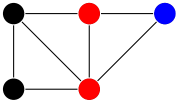
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

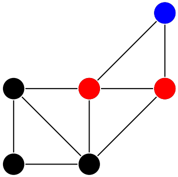
k-trees and partial k-trees

Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

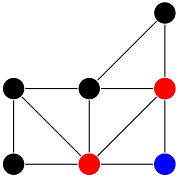
Example of a 2-tree:



A *k-tree* is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a *k-clique*.

k-trees and partial k-trees

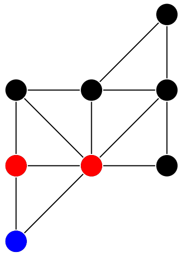
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

k-trees and partial k-trees

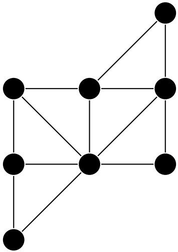
Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

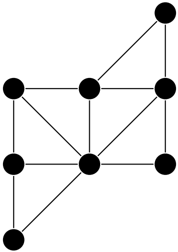
k-trees and partial k-trees

Example of a 2-tree:



A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Example of a 2-tree:



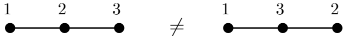
A k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

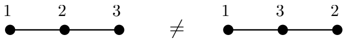
A graph has treewidth at most k if and only if it is a partial k-tree.

What is known about the number of (partial) k-trees?

Labeled k-trees



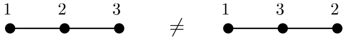
Labeled k-trees



- The number of n-vertex labeled trees is n^{n-2}. [Cayley. 1889]

What is known about the number of (partial) k-trees?

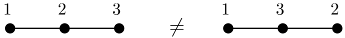
Labeled k-trees



- The number of n-vertex labeled trees is n^{n-2}. [Cayley. 1889]
- The number of n-vertex labeled k-trees is $\binom{n}{k}(kn - k^2 + 1)^{n-k-2}$. [Beineke, Pippert. 1969]

What is known about the number of (partial) k-trees?

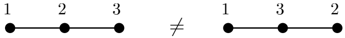
Labeled k-trees



- The number of n-vertex labeled trees is n^{n-2}. [Cayley. 1889]
- The number of n-vertex labeled k-trees is $\binom{n}{k}(kn - k^2 + 1)^{n-k-2}$. [Beineke, Pippert. 1969]

Labeled partial k-trees

Labeled k-trees



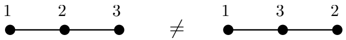
- The number of n-vertex labeled trees is n^{n-2}. [Cayley. 1889]
- The number of n-vertex labeled k-trees is $\binom{n}{k}(kn - k^2 + 1)^{n-k-2}$. [Beineke, Pippert. 1969]

Labeled partial k-trees

- $k = 1$: The number of n-vertex labeled forests is $\sim c \cdot n^{n-2}$ for some constant $c > 1$. [Takács. 1990]

What is known about the number of (partial) k-trees?

Labeled k-trees



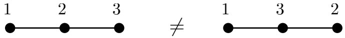
- The number of n-vertex labeled trees is n^{n-2}. [Cayley. 1889]
- The number of n-vertex labeled k-trees is $\binom{n}{k}(kn - k^2 + 1)^{n-k-2}$. [Beineke, Pippert. 1969]

Labeled partial k-trees

- $k = 1$: The number of n-vertex labeled forests is $\sim c \cdot n^{n-2}$ for some constant $c > 1$. [Takács. 1990]
- $k = 2$: The number of n-vertex labeled series-parallel graphs is $\sim g \cdot n^{-\frac{5}{2}}\gamma^n n!$ for some constants $g, \gamma > 0$. [Bodirsky, Giménez, Kang, Noy. 2005]

What is known about the number of (partial) k-trees?

Labeled k-trees



- The number of n-vertex labeled trees is n^{n-2}. [Cayley. 1889]
- The number of n-vertex labeled k-trees is $\binom{n}{k}(kn - k^2 + 1)^{n-k-2}$.
  [Beineke, Pippert. 1969]

Labeled partial k-trees

- $k = 1$: The number of n-vertex labeled forests is $\sim c \cdot n^{n-2}$ for some constant $c > 1$. [Takács. 1990]
- $k = 2$: The number of n-vertex labeled series-parallel graphs is $\sim g \cdot n^{-\frac{5}{2}} \gamma^n n!$ for some constants $g, \gamma > 0$. [Bodirsky, Giménez, Kang, Noy. 2005]
- Nothing was known for general k.

$T_{n,k}$ and an easy upper bound

Let $T_{n,k}$ be the number of n-vertex labeled partial k-trees.

Objective We want to obtain accurate bounds for $T_{n,k}$.

$T_{n,k}$ and an easy upper bound

Let $T_{n,k}$ be the number of n-vertex labeled partial k-trees.

Objective We want to obtain accurate bounds for $T_{n,k}$.

As an n-vertex k-tree has $kn - \frac{k(k+1)}{2}$ edges, we get the upper bound:

$$T_{n,k} \leq \binom{n}{k} \cdot (kn - k^2 + 1)^{n-k-2} \cdot 2^{kn - \frac{k(k+1)}{2}}$$

$T_{n,k}$ and an easy upper bound

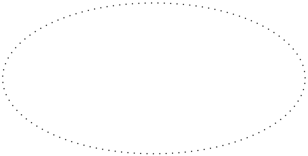
Let $T_{n,k}$ be the number of n-vertex labeled partial k-trees.

Objective We want to obtain accurate bounds for $T_{n,k}$.

As an n-vertex k-tree has $kn - \frac{k(k+1)}{2}$ edges, we get the upper bound:

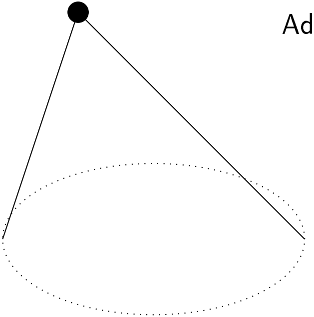
$$
\begin{aligned}
T_{n,k} &\leq \binom{n}{k} \cdot (kn - k^2 + 1)^{n-k-2} \cdot 2^{kn - \frac{k(k+1)}{2}} \\
&\leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}
\end{aligned}
$$

An easy lower bound



Take a forest on $n - (k - 1)$ vertices: $(n - k + 1)^{(n-k-1)}$ possibilities

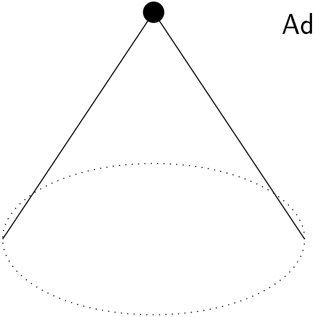
An easy lower bound



Add a vertex arbitrarily connected to the forest:
$2^{n-(k-1)}$ possibilities

Take a forest on $n - (k - 1)$ vertices:
$(n - k + 1)^{(n-k-1)}$ possibilities

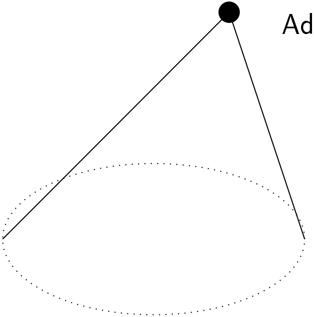
An easy lower bound



Add a vertex arbitrarily connected to the forest:
$2^{n-(k-1)}$ possibilities

Take a forest on $n-(k-1)$ vertices:
$(n-k+1)^{(n-k-1)}$ possibilities

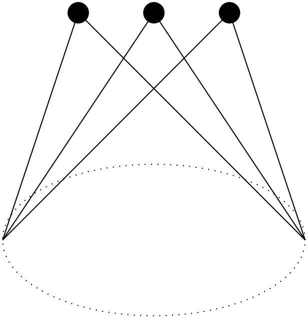
An easy lower bound



Add a vertex arbitrarily connected to the forest:
$2^{n-(k-1)}$ possibilities

Take a forest on $n-(k-1)$ vertices:
$(n-k+1)^{(n-k-1)}$ possibilities

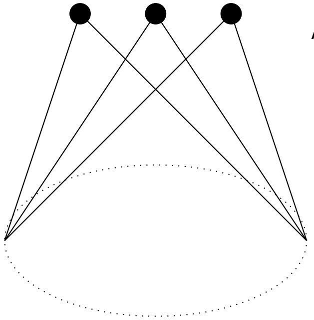
An easy lower bound



Add $k-1$ vertices connected to the forest:
$\geq 2^{(k-1)(n-(k-1))}$ possibilities

Take a forest on $n-(k-1)$ vertices:
$(n-k+1)^{(n-k-1)}$ possibilities

An easy lower bound

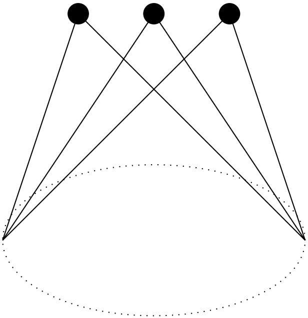


Add $k-1$ vertices connected to the forest:
$\geq 2^{(k-1)(n-(k-1))}$ possibilities

Take a forest on $n-(k-1)$ vertices:
$(n-k+1)^{(n-k-1)}$ possibilities

$$T_{n,k} \geq (n-k+1)^{(n-k-1)} \cdot 2^{(k-1)(n-k+1)}$$

An easy lower bound



Add $k-1$ vertices connected to the forest:
$\geq 2^{(k-1)(n-(k-1))}$ possibilities

Take a forest on $n-(k-1)$ vertices:
$(n-k+1)^{(n-k-1)}$ possibilities

$$T_{n,k} \geq (n-k+1)^{(n-k-1)} \cdot 2^{(k-1)(n-k+1)} \geq \left(\frac{1}{4} \cdot 2^k \cdot n\right)^n \cdot 2^{-k^2}$$

Our results

Summarizing, so far we have:

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}$$

$$T_{n,k} \geq \left(\frac{1}{4} \cdot 2^k \cdot n\right)^n \cdot 2^{-k^2}$$

Our results

Summarizing, so far we have:

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}$$

$$T_{n,k} \geq \left(\frac{1}{4} \cdot 2^k \cdot n\right)^n \cdot 2^{-k^2}$$

Gap in the dominant term: $(4 \cdot k)^n$

Our results

Summarizing, so far we have:

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}$$

$$T_{n,k} \geq \left(\frac{1}{4} \cdot 2^k \cdot n\right)^n \cdot 2^{-k^2}$$

Gap in the dominant term: $(4 \cdot k)^n$

Theorem (Baste, Noy, S.)

For any two integers n, k with $1 < k \leq n$, the number $T_{n,k}$ of n-vertex labeled graphs with treewidth at most k satisfies

$$T_{n,k} \geq \left(\frac{1}{128e} \cdot \frac{k \cdot 2^k \cdot n}{\log k}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

Our results

Summarizing, so far we have:

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}$$

$$T_{n,k} \geq \left(\frac{1}{4} \cdot 2^k \cdot n\right)^n \cdot 2^{-k^2}$$

Gap in the dominant term: $(4 \cdot k)^n$

Theorem (Baste, Noy, S.)

For any two integers n, k with $1 < k \leq n$, the number $T_{n,k}$ of n-vertex labeled graphs with treewidth at most k satisfies

$$T_{n,k} \geq \left(\frac{1}{128e} \cdot \frac{k \cdot 2^k \cdot n}{\log k}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

Gap in the dominant term: $(128e \cdot \log k)^n$

Next section is...

A construction to get a "good" lower bound

Trade-off creating many graphs vs. bounding the number of duplicates

A construction to get a "good" lower bound

Trade-off creating many graphs vs. bounding the number of duplicates

Some ingredients of the construction:

1. labeling function σ: permutation of $\{1, \ldots, n\}$ with $\sigma(1) = 1$.
2. We will introduce vertices $\{v_1, v_2, \ldots, v_n\}$ one by one following the order $v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(n)}$.
3. If $j < i$, the vertex $v_{\sigma(j)}$ is said to be to the left of $v_{\sigma(i)}$.

Proper-pathwidth

Another graph invariant: proper-pathwidth. [Takahashi, Ueno, Kajitani. 1994]

Proper-pathwidth

Another graph invariant: proper-pathwidth. [Takahashi, Ueno, Kajitani. 1994]

Proper linear k-trees: graphs that can be constructed starting from a $(k+1)$-clique and iteratively adding a vertex v_i connected to a clique K_{v_i} of size k (called the active vertices), with the constraints that

- $v_{i-1} \in K_{v_i}$.
- $K_{v_i} \setminus \{v_{i-1}\} \subseteq K_{v_{i-1}}$.

Proper-pathwidth

Another graph invariant: proper-pathwidth. [Takahashi, Ueno, Kajitani. 1994]

Proper linear k-trees: graphs that can be constructed starting from a $(k+1)$-clique and iteratively adding a vertex v_i connected to a clique K_{v_i} of size k (called the active vertices), with the constraints that

- $v_{i-1} \in K_{v_i}$.
- $K_{v_i} \setminus \{v_{i-1}\} \subseteq K_{v_{i-1}}$.

Proper-pathwidth of a graph G, denoted **ppw**(G):
smallest k such that G is a subgraph of a proper linear k-tree.

Another graph invariant: proper-pathwidth. [Takahashi, Ueno, Kajitani. 1994]

Proper linear k-trees: graphs that can be constructed starting from a $(k+1)$-clique and iteratively adding a vertex v_i connected to a clique K_{v_i} of size k (called the active vertices), with the constraints that

- $v_{i-1} \in K_{v_i}$.
- $K_{v_i} \setminus \{v_{i-1}\} \subseteq K_{v_{i-1}}$.

Proper-pathwidth of a graph G, denoted $\mathbf{ppw}(G)$:
smallest k such that G is a subgraph of a proper linear k-tree.

For any graph G it holds that

$$\mathbf{tw}(G) \leq \mathbf{pw}(G) \leq \mathbf{ppw}(G)$$

Proper-pathwidth

Another graph invariant: proper-pathwidth. [Takahashi, Ueno, Kajitani. 1994]

Proper linear k-trees: graphs that can be constructed starting from a $(k+1)$-clique and iteratively adding a vertex v_i connected to a clique K_{v_i} of size k (called the active vertices), with the constraints that

- $v_{i-1} \in K_{v_i}$.
- $K_{v_i} \setminus \{v_{i-1}\} \subseteq K_{v_{i-1}}$.

Proper-pathwidth of a graph G, denoted $\mathbf{ppw}(G)$:
smallest k such that G is a subgraph of a proper linear k-tree.

For any graph G it holds that

$$\mathbf{tw}(G) \leq \mathbf{pw}(G) \leq \mathbf{ppw}(G)$$

The graphs G we will construct satisfy $\mathbf{tw}(G) \leq \mathbf{pw}(G) \leq \mathbf{ppw}(G) \leq k$.

Ingredients of the construction

For every $i \geq k + 1$ we define:

1. A set $A_i \subseteq \{j \mid j < i\}$ with $|A_i| = k$ of active vertices (as in the definition of proper linear k-trees).

Ingredients of the construction

For every $i \geq k+1$ we define:

1. A set $A_i \subseteq \{j \mid j < i\}$ with $|A_i| = k$ of active vertices (as in the definition of proper linear k-trees).
2. A set $N(i) \subseteq A_i$ with $|N(i)| > \frac{k+1}{2}$: neighbors of $v_{\sigma(i)}$ to the left.

Ingredients of the construction

For every $i \geq k+1$ we define:

1. A set $A_i \subseteq \{j \mid j < i\}$ with $|A_i| = k$ of active vertices (as in the definition of proper linear k-trees).
2. A set $N(i) \subseteq A_i$ with $|N(i)| > \frac{k+1}{2}$: neighbors of $v_{\sigma(i)}$ to the left.
3. An element $f(i) \in A_i \cap N(i-1)$, called the forgotten vertex: a vertex that will not be active anymore.

Ingredients of the construction

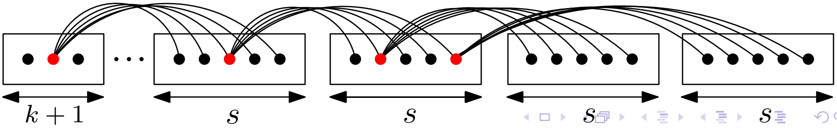
For every $i \geq k+1$ we define:

1. A set $A_i \subseteq \{j \mid j < i\}$ with $|A_i| = k$ of active vertices (as in the definition of proper linear k-trees).
2. A set $N(i) \subseteq A_i$ with $|N(i)| > \frac{k+1}{2}$: neighbors of $v_{\sigma(i)}$ to the left.
3. An element $f(i) \in A_i \cap N(i-1)$, called the forgotten vertex: a vertex that will not be active anymore.
4. We insert the vertices by consecutive blocks of size $s = s(n,k)$. We will fix the value of s later.

Ingredients of the construction

For every $i \geq k+1$ we define:

1. A set $A_i \subseteq \{j \mid j < i\}$ with $|A_i| = k$ of active vertices (as in the definition of proper linear k-trees).
2. A set $N(i) \subseteq A_i$ with $|N(i)| > \frac{k+1}{2}$: neighbors of $v_{\sigma(i)}$ to the left.
3. An element $f(i) \in A_i \cap N(i-1)$, called the forgotten vertex: a vertex that will not be active anymore.
4. We insert the vertices by consecutive blocks of size $s = s(n,k)$. We will fix the value of s later.
5. A vertex $a_i \in A_i$, called the anchor: all vertices of the same block are adjacent to the same anchor a_i.

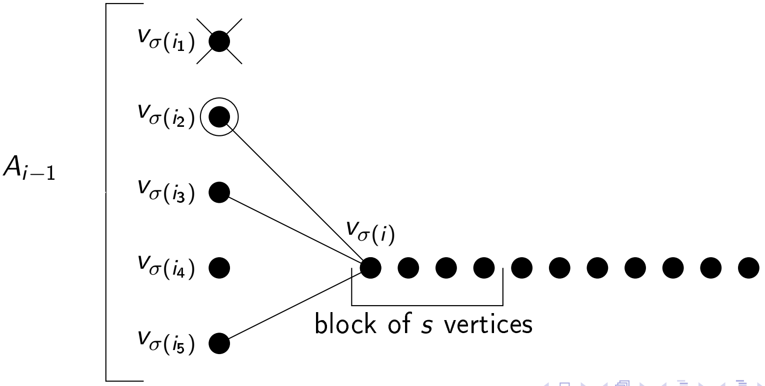


Description of the construction

1. Choose σ, a permutation of $\{1, \dots, n\}$ such that $\sigma(1) = 1$.
2. Choose the first $(k+1)$-clique, with $1 \in N(i)$ for $2 \leq i \leq k+1$.
3. Define $a_{k+1} = 1$.

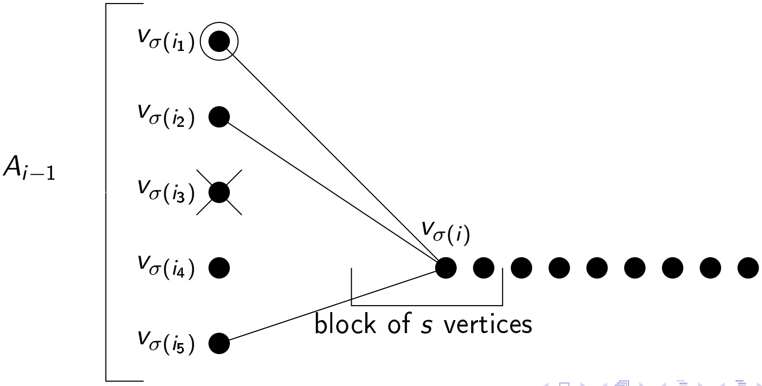
Description of the construction

1. If $i \equiv k+2 \pmod{s}$ (that is, at the beginning of a block):
   - Define $f(i) = a_{i-1}$.
   - Define $A_i = (A_{i-1} \setminus \{f(i)\}) \cup \{i-1\}$.
   - Define $a_i = \min A_i$.
   - Choose $N(i) \subseteq A_i$ such that $a_i \in N(i)$ and $|N(i)| > \frac{k+1}{2}$.

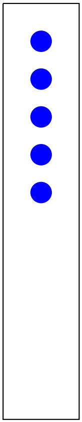


Description of the construction

1. If $i \not\equiv k+2 \pmod{s}$ (that is, at the middle of a block):
   - Choose $f(i) \in (A_{i-1} \setminus \{a_{i-1}\}) \cap N(i-1)$.
   - Define $A_i = (A_{i-1} \setminus \{f(i)\}) \cup \{i-1\}$.
   - Define $a_i = a_{i-1}$.
   - Choose $N(i) \subseteq A_i$ such that $a_i \in N(i)$ and $|N(i)| > \frac{k+1}{2}$.

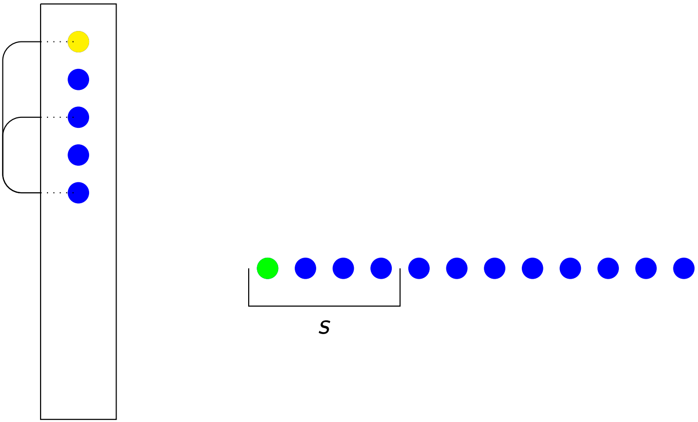


Active vertices



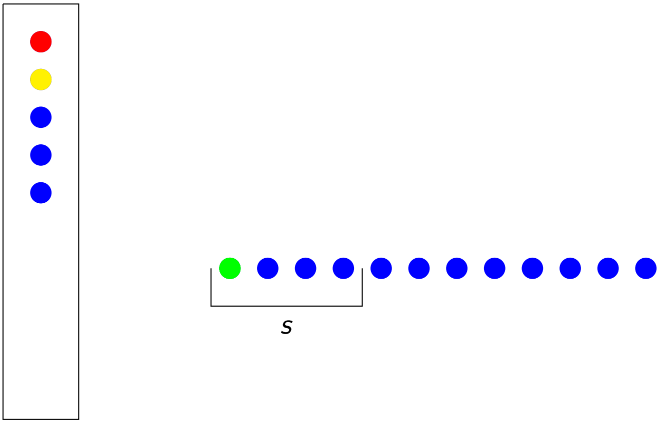
Active vertices

Active vertices



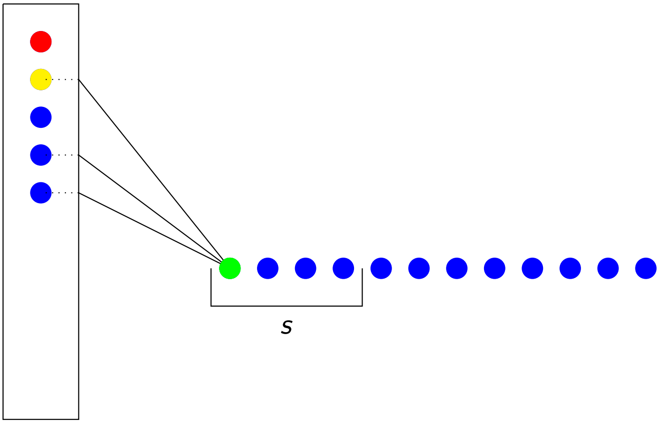
Example of the construction for $k = 4$ (by Julien Baste)

Active vertices

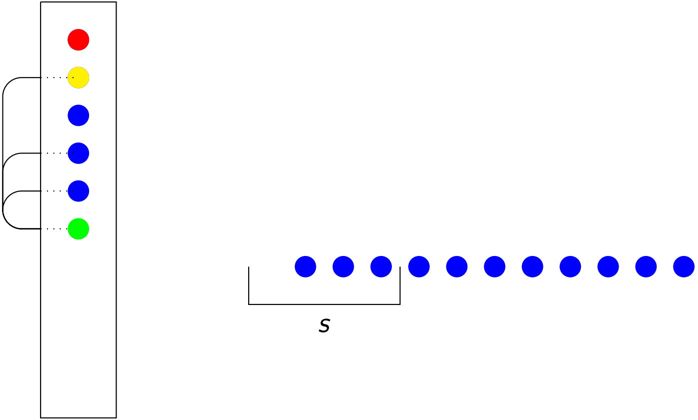


Example of the construction for $k = 4$ (by Julien Baste)

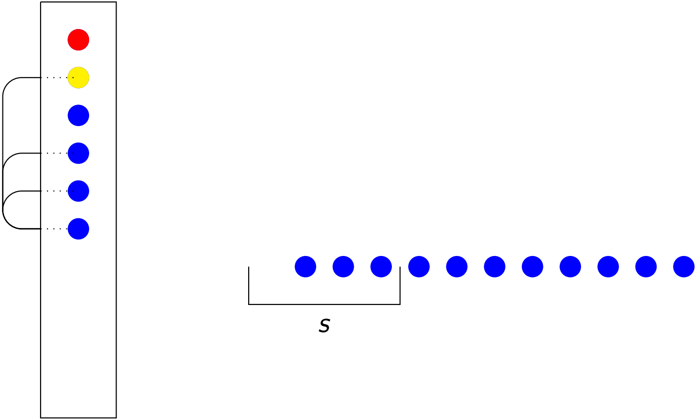
Active vertices



Active vertices

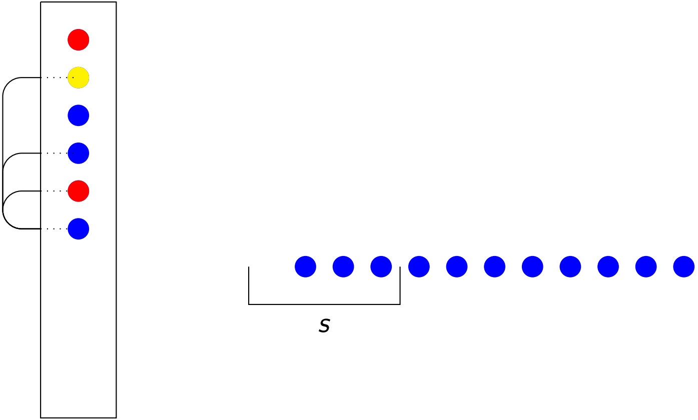


Active vertices



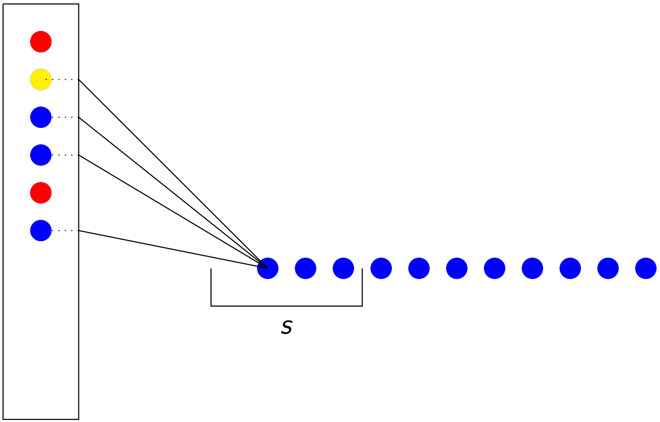
Example of the construction for $k = 4$ (by Julien Baste)

Active vertices

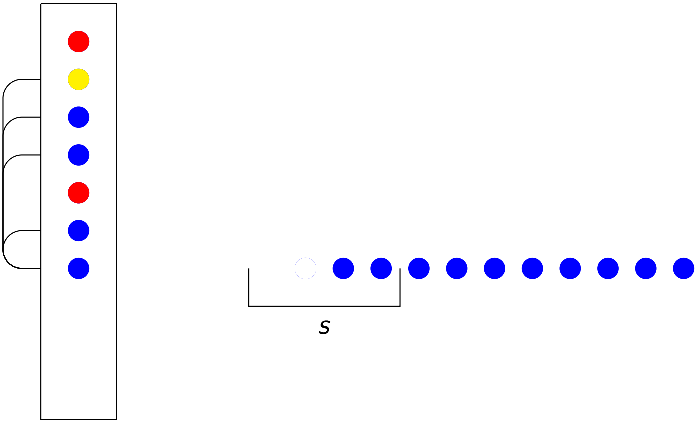


Example of the construction for $k = 4$ (by Julien Baste)

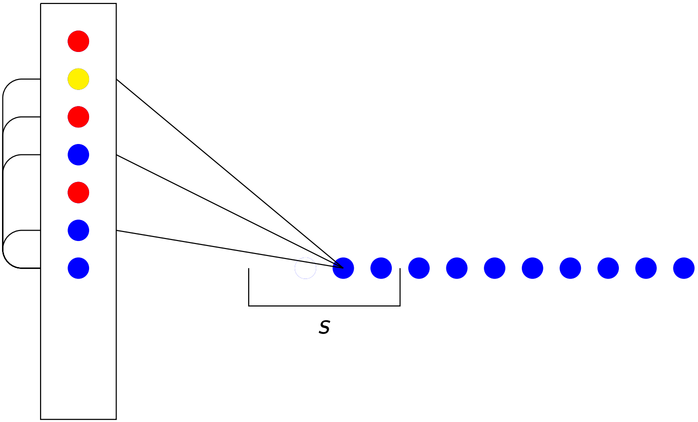
Active vertices



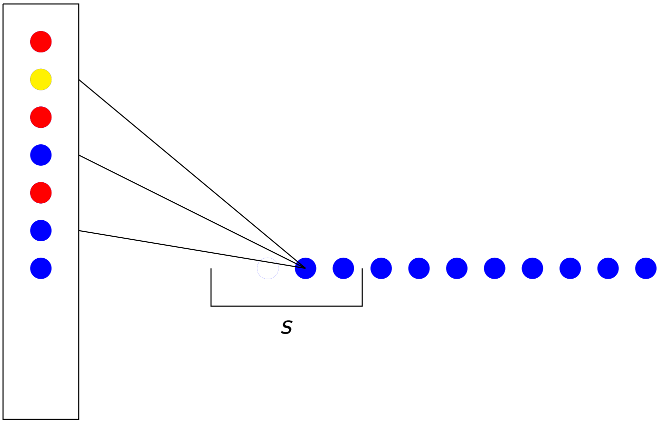
Active vertices



Active vertices

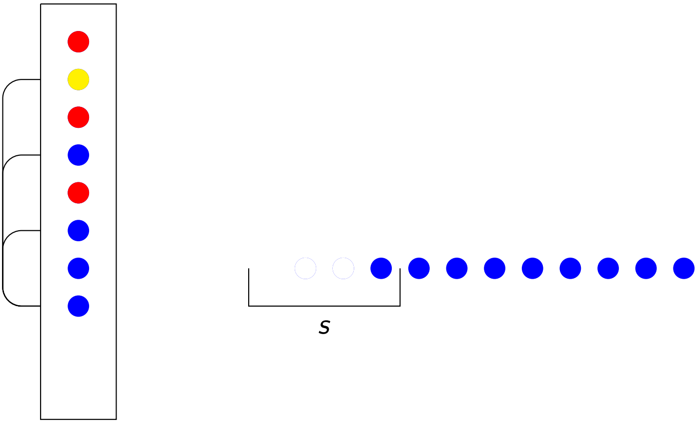


Active vertices



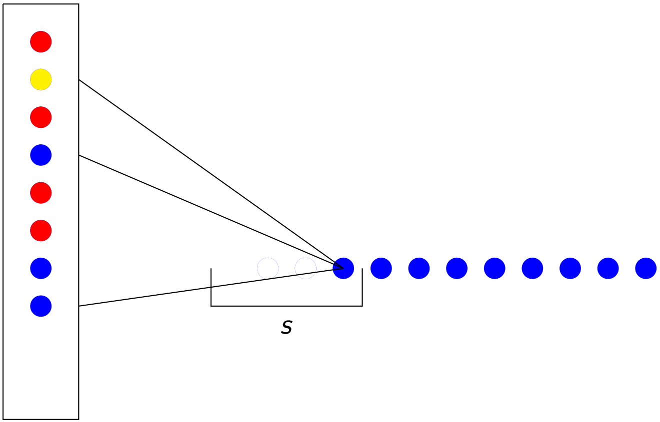
Example of the construction for $k = 4$ (by Julien Baste)

Active vertices

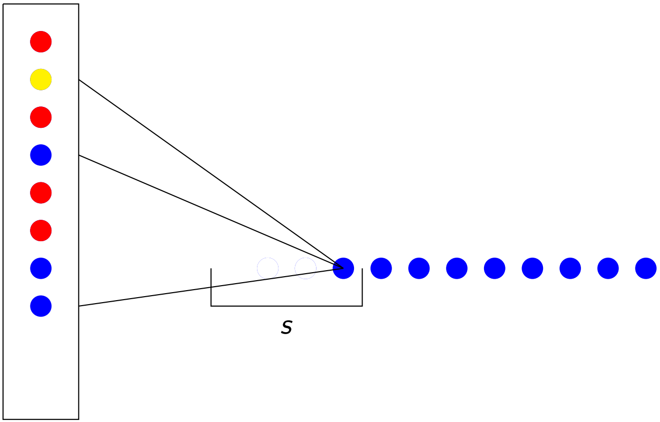


Example of the construction for $k = 4$ (by Julien Baste)

Active vertices

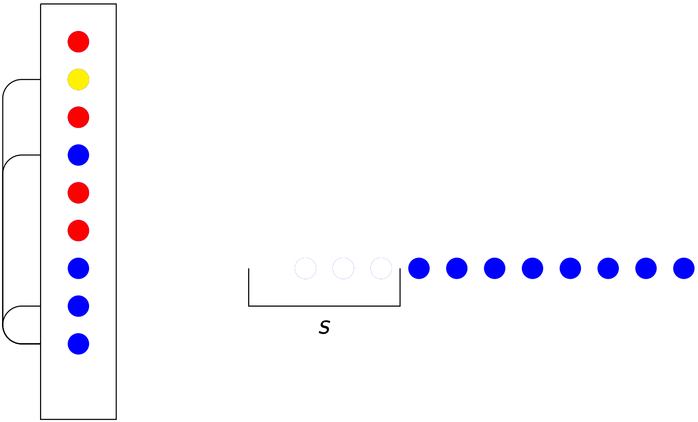


Active vertices



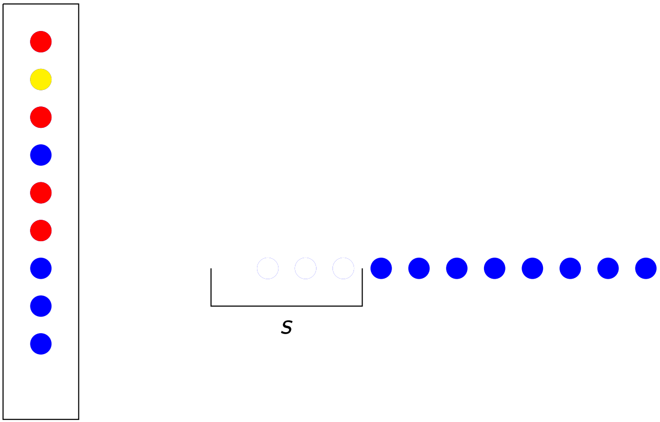
Example of the construction for $k = 4$ (by Julien Baste)

Active vertices

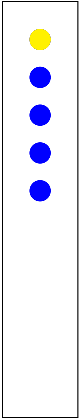


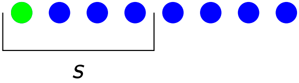
Example of the construction for $k = 4$ (by Julien Baste)

Active vertices



Active vertices





Next section is...

Analysis of the construction

First note that the graphs G we construct indeed satisfy $\mathbf{ppw}(G) \leq k$.

Analysis of the construction

- How many graphs are created by the construction?
- How many times the same graph may have been created?

Analysis of the construction

- How many graphs are created by the construction?

Analysis of the construction

- How many graphs are created by the construction?

The choices in the construction are the following:

- Choices for the permutation σ: $(n-1)!$

Analysis of the construction

- How many graphs are created by the construction?

The choices in the construction are the following:

- Choices for the permutation σ: $(n-1)!$
- Choices for the neighborhoods $N(i)$: $2^{\frac{k(k-1)}{2}} \cdot 2^{(n-(k+1))(k-2)}$

Analysis of the construction

- How many graphs are created by the construction?

The choices in the construction are the following:

- Choices for the permutation σ: $(n-1)!$
- Choices for the neighborhoods $N(i)$: $2^{\frac{k(k-1)}{2}} \cdot 2^{(n-(k+1))(k-2)}$
- Choices for the forgotten vertices $f(i)$: $\left(\frac{k-1}{2}\right)^{(n-(k+1)-\lceil \frac{n-(k+1)}{s} \rceil)}$

Analysis of the construction

- How many graphs are created by the construction?

The choices in the construction are the following:

- Choices for the permutation σ: $(n-1)!$
- Choices for the neighborhoods $N(i)$: $2^{\frac{k(k-1)}{2}} \cdot 2^{(n-(k+1))(k-2)}$
- Choices for the forgotten vertices $f(i)$: $\left(\frac{k-1}{2}\right)^{(n-(k+1)-\lceil\frac{n-(k+1)}{s}\rceil)}$

That is, we create

$$(n-1)! \cdot \left(\frac{k-1}{2}\right)^{(n-(k+1)-\lceil\frac{n-(k+1)}{s}\rceil)} \cdot 2^{\frac{k(k-1)}{2}} \cdot 2^{(n-(k+1))(k-2)}$$

graphs.

How many times the same graph may have been created?

Note that a triple (σ, N, f) uniquely defines a graph $H = G(\sigma, N, f)$.

How many times the same graph may have been created?

Note that a triple (σ, N, f) uniquely defines a graph $H = G(\sigma, N, f)$.

So given an arbitrary constructible graph H, we want to upper-bound the number of triples (σ, N, f) such that $H = G(\sigma, N, f)$.

How many times the same graph may have been created?

Note that a triple (σ, N, f) uniquely defines a graph $H = G(\sigma, N, f)$.

So given an arbitrary constructible graph H, we want to upper-bound the number of triples (σ, N, f) such that $H = G(\sigma, N, f)$.

First we reconstruct the permutation σ:

- $\sigma(1) = 1$ and $f(k+2) = 1$:

How many times the same graph may have been created?

Note that a triple (σ, N, f) uniquely defines a graph $H = G(\sigma, N, f)$.

So given an arbitrary constructible graph H, we want to upper-bound the number of triples (σ, N, f) such that $H = G(\sigma, N, f)$.

First we reconstruct the permutation σ:

- $\sigma(1) = 1$ and $f(k+2) = 1$: images by σ of $\{2, \ldots, k+1\}$ uniquely determined: $k!$ possibilities for ordering the first $k+1$ vertices.

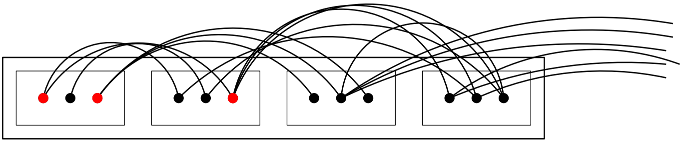
How many times the same graph may have been created?

Note that a triple (σ, N, f) uniquely defines a graph $H = G(\sigma, N, f)$.

So given an arbitrary constructible graph H, we want to upper-bound the number of triples (σ, N, f) such that $H = G(\sigma, N, f)$.

First we reconstruct the permutation σ:

- $\sigma(1) = 1$ and $f(k+2) = 1$: images by σ of $\{2, \ldots, k+1\}$ uniquely determined: $k!$ possibilities for ordering the first $k+1$ vertices.



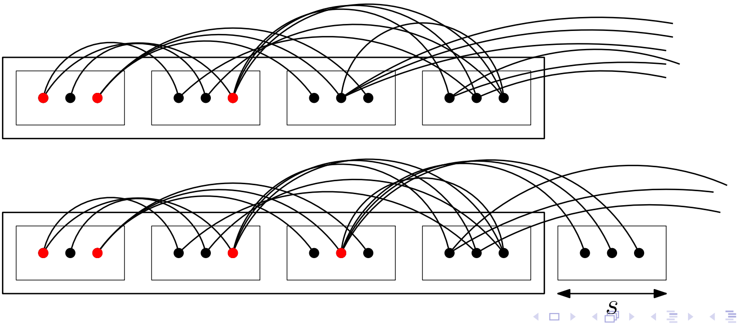
How many times the same graph may have been created?

Note that a triple (σ, N, f) uniquely defines a graph $H = G(\sigma, N, f)$.

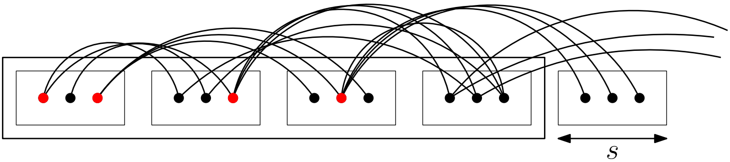
So given an arbitrary constructible graph H, we want to upper-bound the number of triples (σ, N, f) such that $H = G(\sigma, N, f)$.

First we reconstruct the permutation σ:

- $\sigma(1) = 1$ and $f(k+2) = 1$: images by σ of $\{2, \ldots, k+1\}$ uniquely determined: $k!$ possibilities for ordering the first $k+1$ vertices.



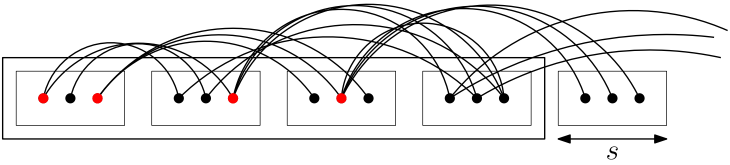
How many times the same graph may have been created?



So the number of possible permutations σ that give rise to H is at most

$$k! \cdot (s!)^{\lceil \frac{n-(k+1)}{s} \rceil}$$

How many times the same graph may have been created?

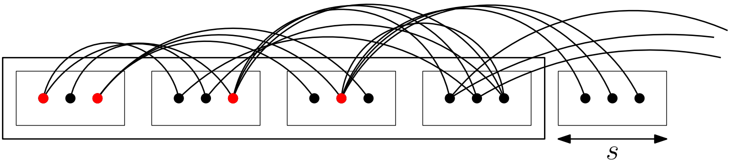


So the number of possible permutations σ that give rise to H is at most

$$k! \cdot (s!)^{\lceil \frac{n-(k+1)}{s} \rceil}$$

Secondly, we reconstruct the neighborhood $N(i)$:

How many times the same graph may have been created?



So the number of possible permutations σ that give rise to H is at most

$$k! \cdot (s!)^{\lceil \frac{n-(k+1)}{s} \rceil}$$

Secondly, we reconstruct the neighborhood $N(i)$:

uniquely determined once σ is fixed.

Reconstruction of the forgotten vertex $f(i)$

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.
- Let $I = \{i \in \{k+1, \ldots, n\} \mid |D_i| \geq 2\}$, and note that $|I| \leq k$.
- It holds that $\sum_{i \in I} |D_i| \leq 2k$.

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.
- Let $I = \{i \in \{k+1, \ldots, n\} \mid |D_i| \geq 2\}$, and note that $|I| \leq k$.
- It holds that $\sum_{i \in I} |D_i| \leq 2k$.

The number of distinct functions f is at most

$$\prod_{i=k+1}^{n} |D_i|$$

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.
- Let $I = \{i \in \{k+1, \ldots, n\} \mid |D_i| \geq 2\}$, and note that $|I| \leq k$.
- It holds that $\sum_{i \in I} |D_i| \leq 2k$.

The number of distinct functions f is at most

$$\prod_{i=k+1}^{n} |D_i| = \prod_{i \in I} |D_i|$$

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.
- Let $I = \{i \in \{k+1, \ldots, n\} \mid |D_i| \geq 2\}$, and note that $|I| \leq k$.
- It holds that $\sum_{i \in I} |D_i| \leq 2k$.

The number of distinct functions f is at most

$$\prod_{i=k+1}^{n} |D_i| = \prod_{i \in I} |D_i| \leq \left(\frac{\sum_{i \in I} |D_i|}{k} \right)^k \leq$$

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.
- Let $I = \{i \in \{k+1, \ldots, n\} \mid |D_i| \geq 2\}$, and note that $|I| \leq k$.
- It holds that $\sum_{i \in I} |D_i| \leq 2k$.

The number of distinct functions f is at most

$$\prod_{i=k+1}^{n} |D_i| = \prod_{i \in I} |D_i| \leq \left(\frac{\sum_{i \in I} |D_i|}{k} \right)^k \leq \left(\frac{2k}{k} \right)^k = 2^k.$$

Reconstruction of the forgotten vertex $f(i)$

We define, for $i > 1$, D_i as the set of neighbors of i that will never have any neighbor among the non-introduced vertices.

- $f(i) \in D_{i-1}$.
- $|D_i| \geq 1$ and $D_i \cap D_j = \emptyset$ for $i \neq j$.
- $\sum_{i=k+1}^{n} |D_i| \leq n$.
- Let $I = \{i \in \{k+1, \ldots, n\} \mid |D_i| \geq 2\}$, and note that $|I| \leq k$.
- It holds that $\sum_{i \in I} |D_i| \leq 2k$.

The number of distinct functions f is at most

$$\prod_{i=k+1}^{n} |D_i| = \prod_{i \in I} |D_i| \leq \left(\frac{\sum_{i \in I} |D_i|}{k} \right)^k \leq \left(\frac{2k}{k} \right)^k = 2^k.$$

So, the number of triples (σ, N, f) such that $H = G(\sigma, N, f)$ is at most

$$2^k \cdot k! \cdot (s!)^{\lceil \frac{n-(k+1)}{s} \rceil}$$

Analysis of the construction

The number of distinct graphs we have created is at least

$$\frac{\text{number of created graphs}}{\text{number of duplicates}}$$

The number of distinct graphs we have created is at least

$$\frac{\text{number of created graphs}}{\text{number of duplicates}}$$

$$\geq \frac{(n-1)! \cdot \left(\frac{k-1}{2}\right)^{n-(k+1)-\lceil \frac{n-(k+1)}{s} \rceil} \cdot 2^{\frac{k(k-1)}{2}} \cdot 2^{(n-(k+1))(k-2)}}{2^k \cdot k! \cdot (s!)^{\lfloor \frac{n-(k+1)}{s} \rfloor} \cdot (n-(k+1)-s\lfloor \frac{n-(k+1)}{s} \rfloor)!}.$$

The number of distinct graphs we have created is at least

$$\frac{\text{number of created graphs}}{\text{number of duplicates}}$$

$$\geq \frac{(n-1)! \cdot \left(\frac{k-1}{2}\right)^{n-(k+1)-\lceil \frac{n-(k+1)}{s} \rceil} \cdot 2^{\frac{k(k-1)}{2}} \cdot 2^{(n-(k+1))(k-2)}}{2^k \cdot k! \cdot (s!)^{\lfloor \frac{n-(k+1)}{s} \rfloor} \cdot (n-(k+1)-s\lfloor \frac{n-(k+1)}{s} \rfloor)!}.$$

$$\geq \ \dots \ \geq \left(\frac{1}{64e} \cdot \frac{k \cdot 2^k \cdot n}{k^{\frac{1}{s}} \cdot s}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

Choice of the block size s

We want the value of $s = s(n, k)$ that minimizes $k^{\frac{1}{s}} \cdot s$.

Choice of the block size s

We want the value of $s = s(n, k)$ that minimizes $k^{\frac{1}{s}} \cdot s$.

With $t(n, k) := \frac{s(n,k)}{\log k}$, we have

Choice of the block size s

We want the value of $s = s(n, k)$ that minimizes $k^{\frac{1}{s}} \cdot s$.

With $t(n, k) := \frac{s(n,k)}{\log k}$, we have

$$\log\left(k^{\frac{1}{s}} \cdot s\right) = \frac{\log k}{s} + \log s$$

$$= \frac{1}{t} + \log t + \log\log k.$$

Choice of the block size s

We want the value of $s = s(n,k)$ that minimizes $k^{\frac{1}{s}} \cdot s$.

With $t(n,k) := \frac{s(n,k)}{\log k}$, we have

$$\log\left(k^{\frac{1}{s}} \cdot s\right) = \frac{\log k}{s} + \log s$$

$$= \frac{1}{t} + \log t + \log\log k.$$

And the minimum of $\frac{1}{t(n,k)} + \log t(n,k)$ is reached for $t(n,k) = 1$.

Choice of the block size s

We want the value of $s = s(n, k)$ that minimizes $k^{\frac{1}{s}} \cdot s$.

With $t(n, k) := \frac{s(n,k)}{\log k}$, we have

$$\log\left(k^{\frac{1}{s}} \cdot s\right) = \frac{\log k}{s} + \log s$$

$$= \frac{1}{t} + \log t + \log\log k.$$

And the minimum of $\frac{1}{t(n,k)} + \log t(n, k)$ is reached for $t(n, k) = 1$.

So $s(n, k) = \log k$ is the best choice for the block size.

Next section is...

Further research

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}.$$

$$T_{n,k} \geq \left(\frac{1}{128e} \cdot \frac{k \cdot 2^k \cdot n}{\log k}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

Further research

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}.$$

$$T_{n,k} \geq \left(\frac{1}{128e} \cdot \frac{k \cdot 2^k \cdot n}{\log k}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

- We believe that there exist an absolute constant $c > 0$ and a function $f(k)$, with $k^{-2k-2} \leq f(k) \leq k^{-k}$ for every $k > 0$, such that

$$T_{n,k} \geq (c \cdot k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot f(k).$$

Further research

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}.$$

$$T_{n,k} \geq \left(\frac{1}{128e} \cdot \frac{k \cdot 2^k \cdot n}{\log k}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

- We believe that there exist an absolute constant $c > 0$ and a function $f(k)$, with $k^{-2k-2} \leq f(k) \leq k^{-k}$ for every $k > 0$, such that

$$T_{n,k} \geq (c \cdot k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot f(k).$$

- Improve the upper bound for pathwidth or proper-pathwidth?

Further research

$$T_{n,k} \leq (k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot k^{-k}.$$

$$T_{n,k} \geq \left(\frac{1}{128e} \cdot \frac{k \cdot 2^k \cdot n}{\log k}\right)^n \cdot 2^{-\frac{k(k+3)}{2}} \cdot k^{-2k-2}.$$

- We believe that there exist an absolute constant $c > 0$ and a function $f(k)$, with $k^{-2k-2} \leq f(k) \leq k^{-k}$ for every $k > 0$, such that

$$T_{n,k} \geq (c \cdot k \cdot 2^k \cdot n)^n \cdot 2^{-\frac{k(k+1)}{2}} \cdot f(k).$$

- Improve the upper bound for pathwidth or proper-pathwidth?
- Other relevant parameters: branchwidth, cliquewidth, rankwidth, tree-cutwidth, booleanwidth, ...

Gràcies!