Constructive enumeration and uniform random sampling of DAGs

Antoine Genitrini[1] > **Martin Pépin**[1] Alfredo Viola[2]

Work accepted for publication in the proceedings of LAGOS 2021

March 15, 2021

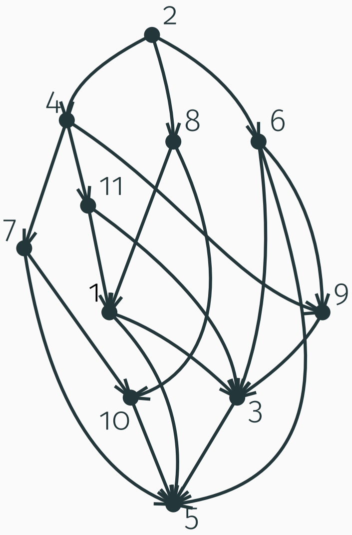
[1]LIP6 — Sorbonne Université — Paris

[2]Universidad de la República — Montevideo

Outline

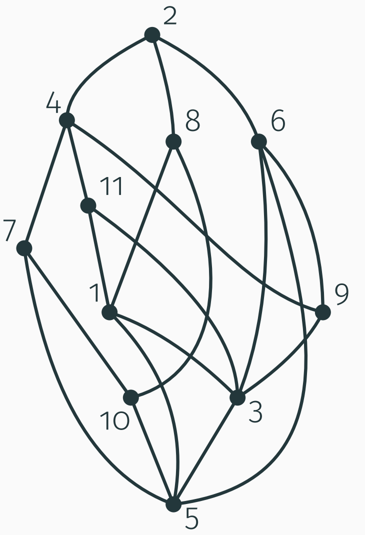
Directed Acyclic Graphs

- A finite set of vertices V e.g. $\{1, 2, \ldots, n\}$;
- a set of directed edges $E \subseteq V \times V$;
- no cycles: $v_1 \to v_2 \to \cdots \to v_n = v_1$.



Directed Acyclic Graphs

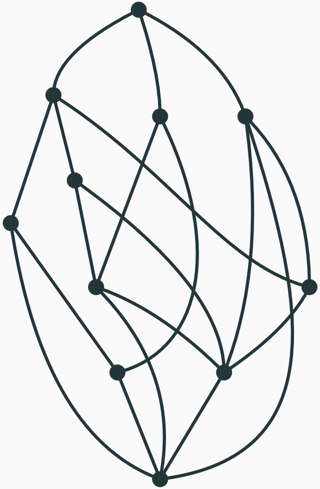
- A finite set of vertices V e.g. $\{1, 2, \ldots, n\}$;
- a set of directed edges $E \subseteq V \times V$;
- no cycles: $v_1 \to v_2 \to \cdots \to v_n = v_1$.



Directed Acyclic Graphs

- A finite set of vertices V e.g. $\{1, 2, \ldots, n\}$;
- a set of directed edges $E \subseteq V \times V$;
- no cycles: $v_1 \to v_2 \to \cdots \to v_n = v_1$.

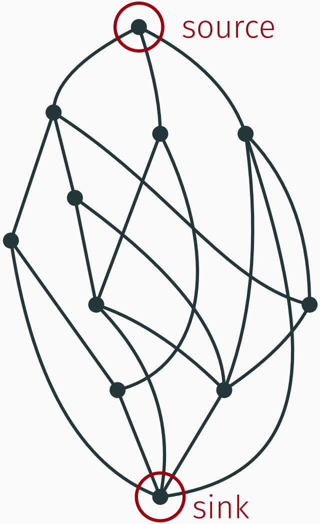
- If considered up to relabelling:
  unlabelled DAGs



Directed Acyclic Graphs

- A finite set of vertices V e.g. $\{1, 2, \ldots, n\}$;
- a set of directed edges $E \subseteq V \times V$;
- no cycles: $v_1 \to v_2 \to \cdots \to v_n = v_1$.

- If considered up to relabelling: **unlabelled DAGs**



State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]

State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]
- Counting by number of edges: [Ges96]

State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]
- Counting by number of edges: [Ges96]
- Uniform sampling: [MDB01], [KM15]

State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]
- Counting by number of edges: [Ges96]
- Uniform sampling: [MDB01], [KM15]

Unlabelled DAGs:

- Counting by vertices and sources: [Rob77]

State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]
- Counting by number of edges: [Ges96]
- Uniform sampling: [MDB01], [KM15]

Unlabelled DAGs:

- Counting by vertices and sources: [Rob77]

Asymptotics

- Cf. Sergey's talk

State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]
- Counting by number of edges: [Ges96] •
- Uniform sampling: [MDB01], [KM15]

Unlabelled DAGs:

- Counting by vertices and sources: [Rob77] •

Asymptotics

- Cf. Sergey's talk

Problems:

- Inclusion-exclusion

State of the art

Labelled DAGs:

- Counting by number of vertices: [Rob73]
- Counting by number of edges: [Ges96] ●
- Uniform sampling: [MDB01], [KM15] ●

Unlabelled DAGs:

- Counting by vertices and sources: [Rob77] ●

Asymptotics

- Cf. Sergey's talk

Problems:

- ● Inclusion-exclusion
- ● No or little control over the number of edges

Still missing

- Finer control over the number of edges?
- Sampling of unlabelled structures?

Outline

Background

Directed Ordered Acyclic Graphs

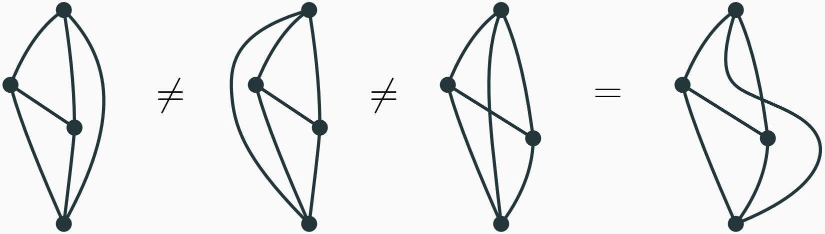
Extensions

Conclusion

A new kind of DAG

Directed Ordered Acyclic Graphs (DOAGs)

DOAG = Unlabelled DAG
+ a total order on the **outgoing** edges of each vertex
+ only one sink and one source



Motivation

- Real-life implementations of DAGs have an **ordering**; →

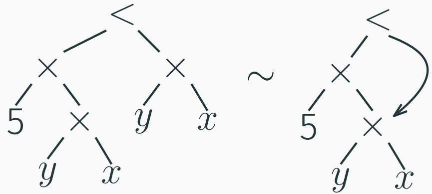
```
struct vertex {
  int                out_degree;
  struct vertex *out_edges;
};
```

Motivation

- Real-life implementations of DAGs have an **ordering**; →

```
struct vertex {
  int           out_degree;
  struct vertex *out_edges;
};
```

- The memory layout of trees with hash-consing have an **ordering**; →

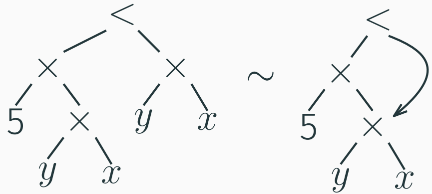


Motivation

- Real-life implementations of DAGs have an **ordering**; →

```
struct vertex {
  int                out_degree;
  struct vertex *out_edges;
};
```

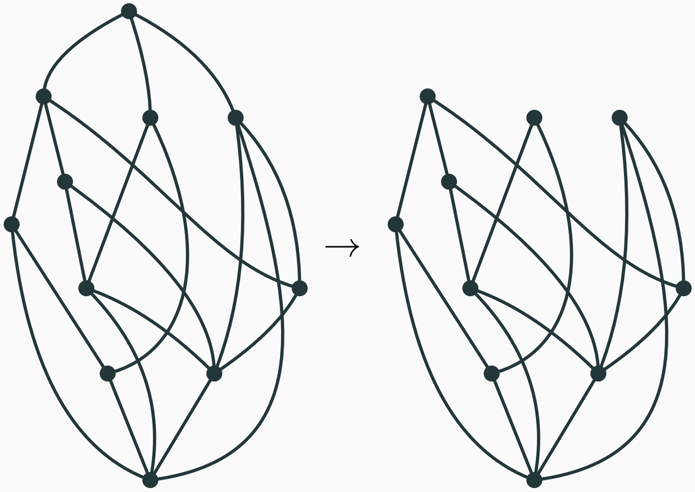
- The memory layout of trees with hash-consing have an **ordering**; →



- 

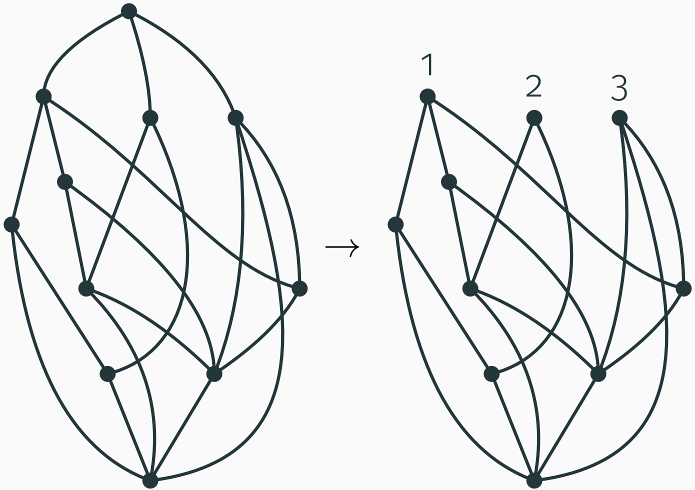
Recursive decomposition: multi-source DOAGs

Idea: remove the source and see what is left.



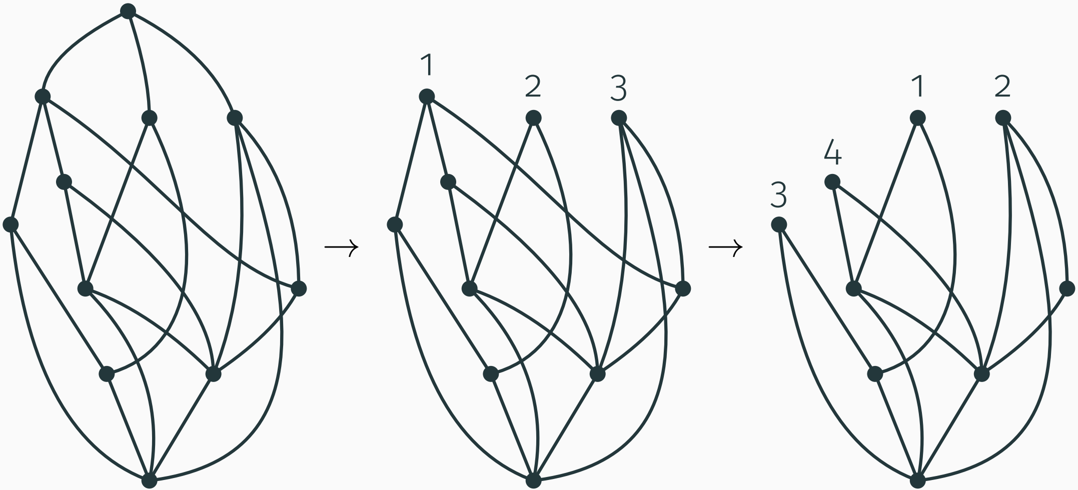
Recursive decomposition: multi-source DOAGs

Idea: remove the source and see what is left.

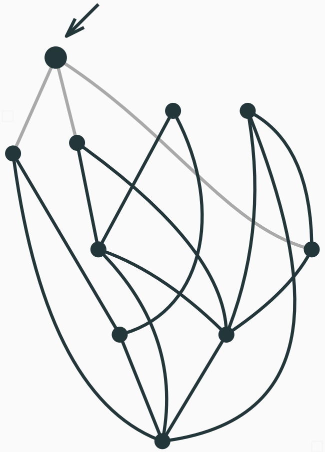


Recursive decomposition: multi-source DOAGs

Idea: remove the source and see what is left.

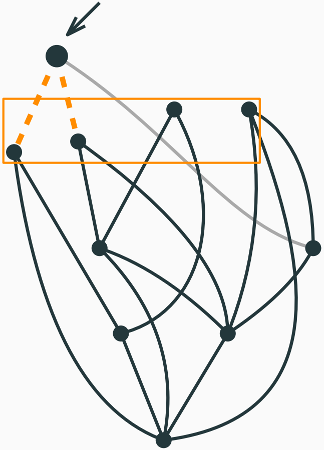


Recursive decomposition: multi-source DOAGs



The smallest source + a sub-DOAG;

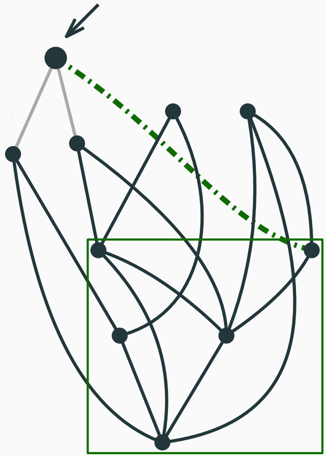
Recursive decomposition: multi-source DOAGs



The smallest source + a sub-DOAG;

q edges to sources;

Recursive decomposition: multi-source DOAGs

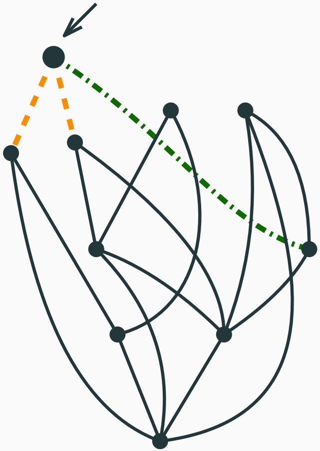


The smallest source + a sub-DOAG;

q edges to **sources**;

s edges to **internal vertices**;

Recursive decomposition: multi-source DOAGs



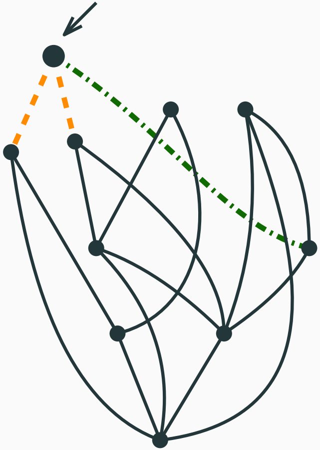
The smallest source + a sub-DOAG;

q edges to **sources**;

s edges to **internal vertices**;

$\binom{s+q}{q} s!$ ways to arrange the two sets of edges;

Recursive decomposition: multi-source DOAGs



The smallest source + a sub-DOAG;

q edges to **sources**;

s edges to **internal vertices**;

$\binom{s+q}{q} s!$ ways to arrange the two sets of edges;

$$D_{n,m,k} = \#\text{DOAGs with } n \text{ vertices, } m \text{ edges, } k \text{ sources}$$
$$= \sum_{s+q>0} D_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s!$$

Complexity of the counting

$$D_{1,m,k} = \mathbb{1}_{\{m=0 \wedge k=1\}}$$

$$D_{n,m,k} = 0 \qquad \text{when } k \leq 0$$

$$D_{n,m,k} = \sum_{s+q>0} D_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s! \qquad \text{when } n > 1$$

Complexity of the counting

$$D_{1,m,k} = \mathbb{1}_{\{m=0 \wedge k=1\}}$$

$$D_{n,m,k} = 0 \qquad \text{when } k \leq 0$$

$$D_{n,m,k} = \sum_{s+q>0} D_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s! \qquad \text{when } n > 1$$

Complexity

Computing $D_{n,m,k}$ for all $n, k \leq N$ and $m \leq M$ takes $O(N^4 M)$ arithmetic operations.

Complexity of the counting

$$
\begin{aligned}
D_{1,m,k} &= \mathbb{1}_{\{m=0 \wedge k=1\}} & \\
D_{n,m,k} &= 0 & \text{when } k \leq 0 \\
D_{n,m,k} &= \sum_{s+q>0} D_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s! & \text{when } n > 1
\end{aligned}
$$

Complexity

Computing $D_{n,m,k}$ for all $n, k \leq N$ and $m \leq M$ takes $O(N^4 M)$ arithmetic operations.

In practice we reach $M = 400$, $N = M + 1$.

Random sampling = GИITИUOƆ

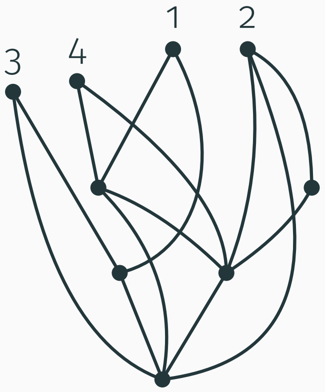
Do the same, but backwards!

Random sampling = ƏИITИUOƆ

Do the same, but backwards!

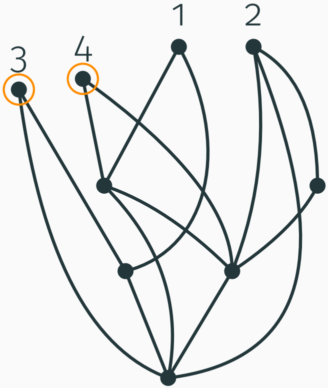
1. Select (s, q) with probability $\frac{D_{n-1,m-s-q,k-1+q}\binom{n-k-q}{s}\binom{s+q}{q}s!}{D_{n,m,k}}$;

Do the same, but backwards!



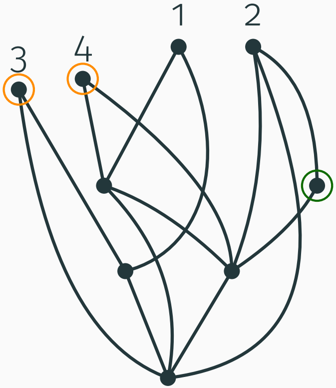
1. Select (s, q) with probability $\frac{D_{n-1,m-s-q,k-1+q}\binom{n-k-q}{s}\binom{s+q}{q}s!}{D_{n,m,k}}$;
2. Sample a DOAG$(n-1, m-s-q, k-1+q)$;

Do the same, but backwards!



1. Select (s, q) with probability $\frac{D_{n-1,m-s-q,k-1+q}\binom{n-k-q}{s}\binom{s+q}{q}s!}{D_{n,m,k}}$;
2. Sample a DOAG$(n-1, m-s-q, k-1+q)$;
3. We already know the q largest sources;

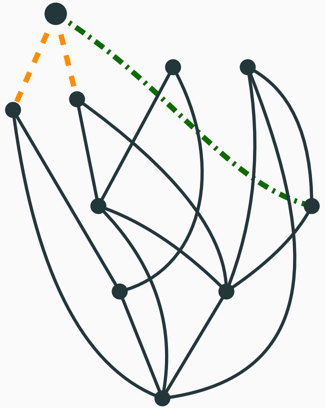
Do the same, but backwards!



1. Select (s, q) with probability $\frac{D_{n-1,m-s-q,k-1+q}\binom{n-k-q}{s}\binom{s+q}{q}s!}{D_{n,m,k}}$;
2. Sample a DOAG$(n-1, m-s-q, k-1+q)$;
3. We already know the q largest sources;
4. Choose s internal vertices;

Random sampling = GИITИUOƆ

Do the same, but backwards!



1. Select (s, q) with probability $\frac{D_{n-1,m-s-q,k-1+q}\binom{n-k-q}{s}\binom{s+q}{q}s!}{D_{n,m,k}}$;
2. Sample a DOAG$(n-1, m-s-q, k-1+q)$;
3. We already know the q largest sources;
4. Choose s internal vertices;
5. Connect them to the new sources.

Complexity of the sampling algorithm

- Selecting s and q: $O((s+q)^2)$ arithmetic operations;
- the rest is cheap.

Complexity of the sampling algorithm

- Selecting s and q: $O((s+q)^2)$ arithmetic operations;
- the rest is cheap.

Complexity

$$\#\ \text{arithmetic operations} = O\left(\sum_{v \text{ vertex}} d_v^2\right)$$

where d_v = out-degree of v.

Complexity of the sampling algorithm

- Selecting s and q: $O((s+q)^2)$ arithmetic operations;
- the rest is cheap.

Complexity

$$\#\text{ arithmetic operations} = O\left(\sum_{v\text{ vertex}} d_v^2\right)$$

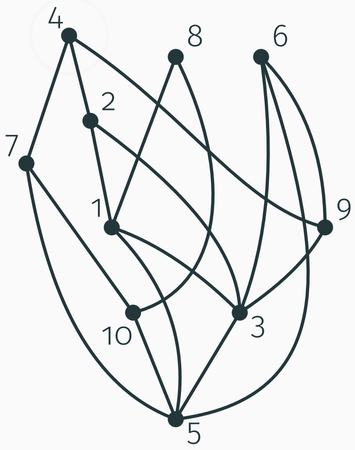
where d_v = out-degree of v.

In practice: a few milliseconds.

Outline

What about labelled DAGs?

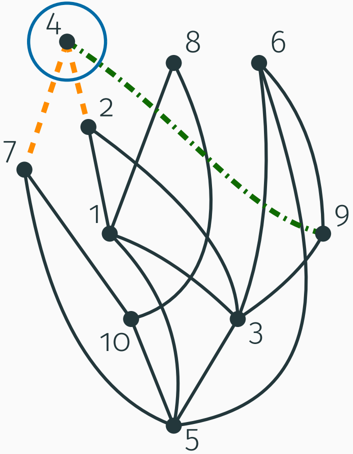
Idea: mark one source, and remove it.



$V_{n,m,k} = \#\text{DAGs (one sink, } k \text{ sources)}$

What about labelled DAGs?

Idea: mark one source, and remove it.

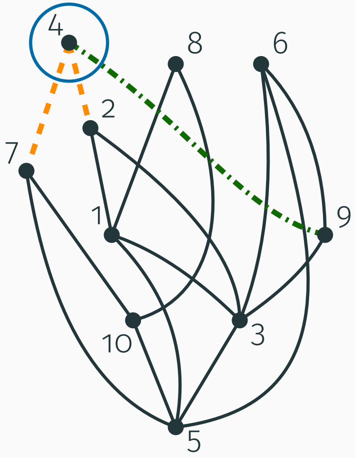


$V_{n,m,k} = \#\text{DAGs (one sink, } k \text{ sources)}$

$k \cdot V_{n,m,k} =$

What about labelled DAGs?

Idea: mark one source, and remove it.



$V_{n,m,k} = \#\text{DAGs (one sink, } k \text{ sources)}$

$k \cdot V_{n,m,k} =$

$$n \cdot \sum_{s+q>0} V_{n-1,m-s-q,k-1+q} \binom{k-1+q}{q} \binom{n-q-k}{s}$$

Bounded degree sampling

What if we want DOAGs with out-degree $\leq d$?

Bounded degree sampling

What if we want DOAGs with out-degree $\leq d$?

$$D_{n,m,k} = \sum_{0<s+q} D_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s!$$

Bounded degree sampling

What if we want DOAGs with out-degree $\leq d$?

$$D^{(d)}_{n,m,k} = \sum_{0<s+q\leq d} D^{(d)}_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s!$$

Bounded degree sampling

What if we want DOAGs with out-degree $\leq d$?

$$D^{(d)}_{n,m,k} = \sum_{0<s+q\leq d} D^{(d)}_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s!$$

- Counting: $O(N^2 d^4)$

Bounded degree sampling

What if we want DOAGs with out-degree $\leq d$?

$$D^{(d)}_{n,m,k} = \sum_{0<s+q\leq d} D^{(d)}_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s!$$

- Counting: $O(N^2 d^4)$
- Sampling: $O(N d^2)$

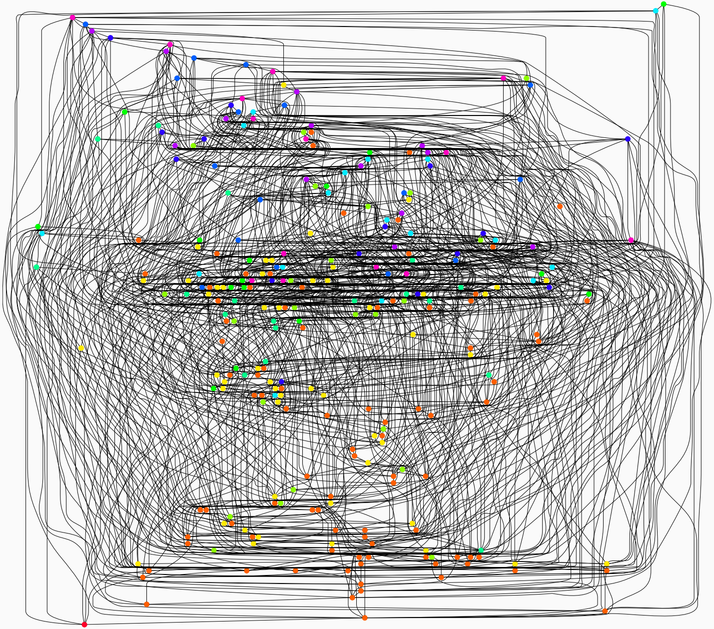
Bounded degree sampling

What if we want DOAGs with out-degree $\leq d$?

$$D^{(d)}_{n,m,k} = \sum_{0<s+q\leq d} D^{(d)}_{n-1,m-s-q,k-1+q} \binom{n-k-q}{s} \binom{s+q}{q} s!$$

- Counting: $O(N^2 d^4)$
- Sampling: $O(N d^2)$
- In practice we reached $m = 1500$ with $d = 2$ and $m = 1000$ with $d = 10$.

Your next favourite wallpaper



Uniform DOAG with $m = 1000$ and $d = 10$.

Outline

Conclusion

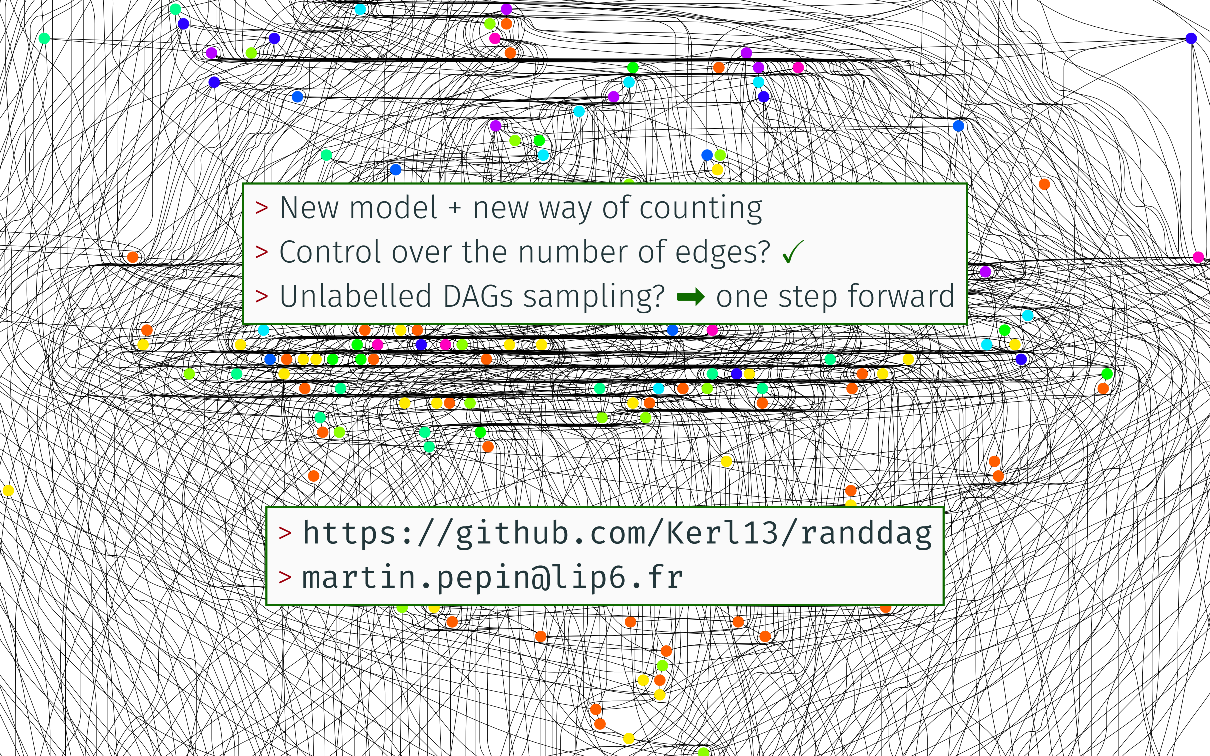
- New model + new way of counting
- Control over the number of edges?
- Unlabelled DAGs sampling?

Conclusion

- New model + new way of counting
- Control over the number of edges? ✓
- Unlabelled DAGs sampling?

Conclusion

- New model + new way of counting
- Control over the number of edges? ✓
- Unlabelled DAGs sampling? ➡ one step forward



- New model + new way of counting
- Control over the number of edges? ✓
- Unlabelled DAGs sampling? ➡ one step forward

- https://github.com/Kerl13/randdag
- martin.pepin@lip6.fr

References i

I. M. Gessel.
Counting acyclic digraphs by sources and sinks.
Discrete Mathematics, 160(1):253 – 258, 1996.

J. Kuipers and G. Moffa.
Uniform random generation of large acyclic digraphs.
Stat. and Computing, 25(2):227–242, 2015.

G. Melançon, I. Dutour, and M. Bousquet-Mélou.
Random generation of directed acyclic graphs.
Electron. Notes Discret. Math., 10:202–207, 2001.

R.W. Robinson.
Counting labeled acyclic digraphs.
New Directions in the Theory of Graphs, pages 239–273, 1973.

R. W. Robinson.
Counting unlabeled acyclic digraphs.
In *Combinatorial Mathematics V*, Lecture Notes in Mathematics, pages 28–43. Springer, 1977.