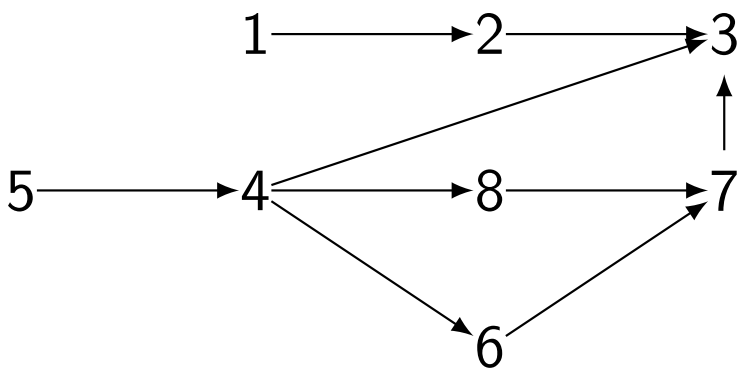
A Comparison of Random Task Graph Generation Methods for Scheduling Problems

Louis-Claude Canon, Mohamad El Sayah
and Pierre-Cyrille Héam

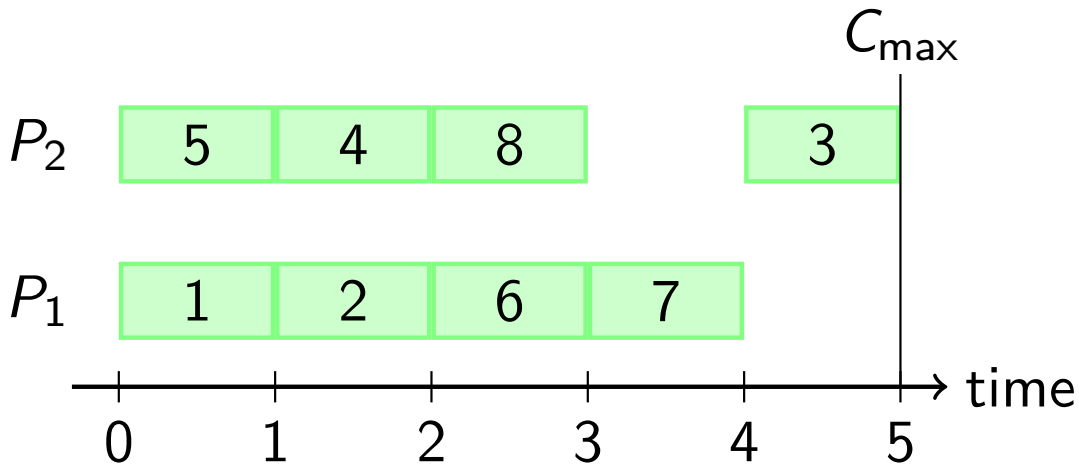
FEMTO-ST Institute, CNRS, Univ. Bourgogne Franche-Comté, France

June 26, 2019

Directed Acyclic Graph (DAG)



$P|p_j = 1, prec|C_{max}$



Random Generation

Validation Approaches:

- Formal analysis
- Simulation on actual traces
- Experimentation in real environments
- Emulation
- Validation on random instances

Random generation:

- Allows comparing strategies on many diverse situations
- May be biased (instances that are not representative of the whole space)

Random Generation

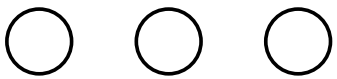
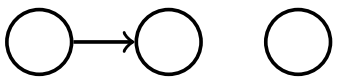
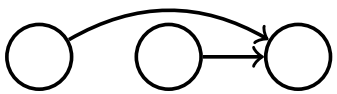
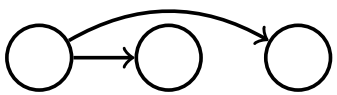
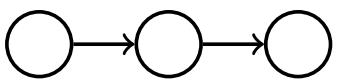
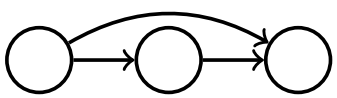
Validation Approaches:

- Formal analysis
- Simulation on actual traces
- Experimentation in real environments
- Emulation
- Validation on random instances

Random generation:

- Allows comparing strategies on many diverse situations
- May be biased (instances that are not representative of the whole space)

Uniformity

Isom. classes	Matrices	ER	Labeling
	$\begin{pmatrix} 0 & 0 \\ & 0 \end{pmatrix}$	$\frac{1}{8}$	1
	$\begin{pmatrix} 1 & 0 \\ & 0 \end{pmatrix}$ $\begin{pmatrix} 0 & 1 \\ & 0 \end{pmatrix}$ $\begin{pmatrix} 0 & 0 \\ & 1 \end{pmatrix}$	$\frac{3}{8}$	6
	$\begin{pmatrix} 0 & 1 \\ & 1 \end{pmatrix}$	$\frac{1}{8}$	3
	$\begin{pmatrix} 1 & 1 \\ & 0 \end{pmatrix}$	$\frac{1}{8}$	3
	$\begin{pmatrix} 1 & 0 \\ & 1 \end{pmatrix}$	$\frac{1}{8}$	6
	$\begin{pmatrix} 1 & 1 \\ & 1 \end{pmatrix}$	$\frac{1}{8}$	6

Related Work

Methods	uniform generation (1973), layer-by-layer (1974), etc.
Real-cases	LU/QR/Cholesky decomposition, parallel Gaussian elimination algorithm, parallel Laplace equation, etc.
Data sets	PSPLIB (1995), TGFF (1998), STG (2002).
Tools	DAGGEN (2009), GGen (2010), Pegasus workflow generator (2013), etc.

Existing Generation Methods

We focus on four methods:

Uniform	Uniform generation among the labeled DAGs
Erdős-Rényi	Uniform generation among the adjacency matrices
Random orders	Combination of random orders
Layer-by-Layer	Ah hoc methods enforcing “layers”

Sub-Exponential Algorithm (Uniform Instances)

- Decompose an instance of size n into many small sub-problems (m)
- Apply exponential brute force approach to solve each sub-problems ($O(2^m)$)
- If each sub-problem size is small ($m = O(\log(n))$), then the strategy is polynomial
- If each sub-problem size is "quite" small ($m = O(\log^k(n))$), then the strategy is sub-exponential

Shape

Shape decomposition $(X_1, X_2, \ldots, X_k)$: X_i contains all vertices of depth i.

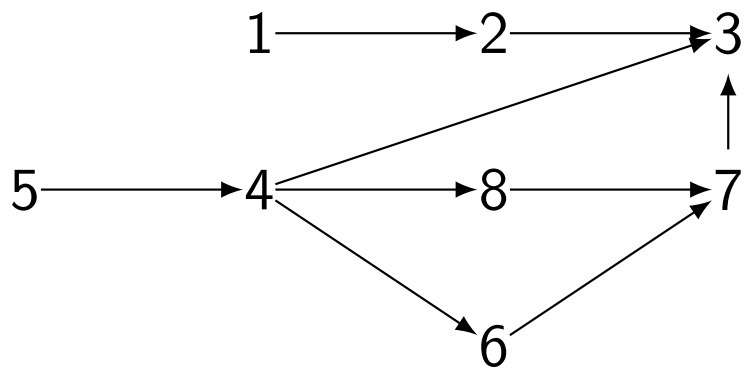
Example

$(\{1,5\},\{2,4\},\{8,6\},\{7\},\{3\})$

Shape

Shape decomposition $(X_1, X_2, \ldots, X_k)$: X_i contains all vertices of depth i.

Example



$(\{1,5\},\{2,4\},\{8,6\},\{7\},\{3\})$

Mass

If $|X_i| = 1$, then all predecessors can be scheduled independently of the successors.
The (absolute) *mass* is the maximum number of vertices that cannot be decomposed (the size of the largest sub-problem).

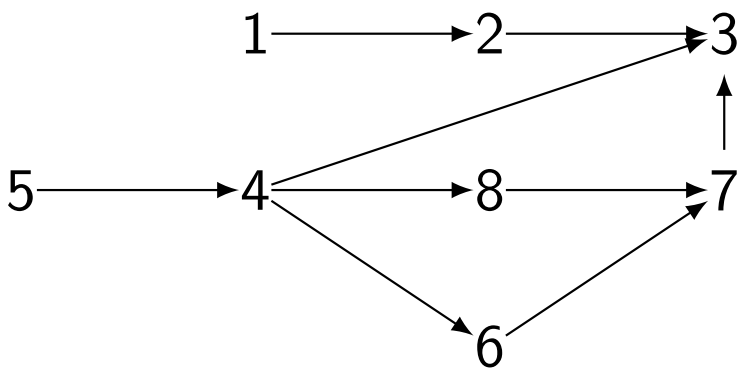
Example

The mass is 6.

Mass

If $|X_i| = 1$, then all predecessors can be scheduled independently of the successors.
The (absolute) *mass* is the maximum number of vertices that cannot be decomposed (the size of the largest sub-problem).

Example



The mass is 6.

Generic-case Complexity

Complexity on most inputs.

Theorem

Let D be a DAG uniformly and randomly generated among the labeled DAGs with n vertices. One has
$\Pr(\text{mass}(D) \geq \log^4(n)) \to 0$ *when* $n \to +\infty$.

Length and Width of Largest Sub-Problem

Length (shape length):

Lemma

For every $n > 0$, $\Pr(\ell_{\max} \geq \ell) \leq n(1-\alpha_0)^{\ell}$.

Width (maximum shape):

Proposition

One has $E(\max(x_i)) = O(\log n)$.

Uniformity over a Subspace

Uniform generation among instances with maximum mass ($= n$).
Which other characteristics should be constrained?

DAG Properties

- number of vertices
- number of edges
- maximum/minimum/mean/sd out/in/total degree
- length
- width
- maximum/minimum/mean/sd shape
- equivalent properties on the reverse shape
- equivalent properties on the transitive reduction

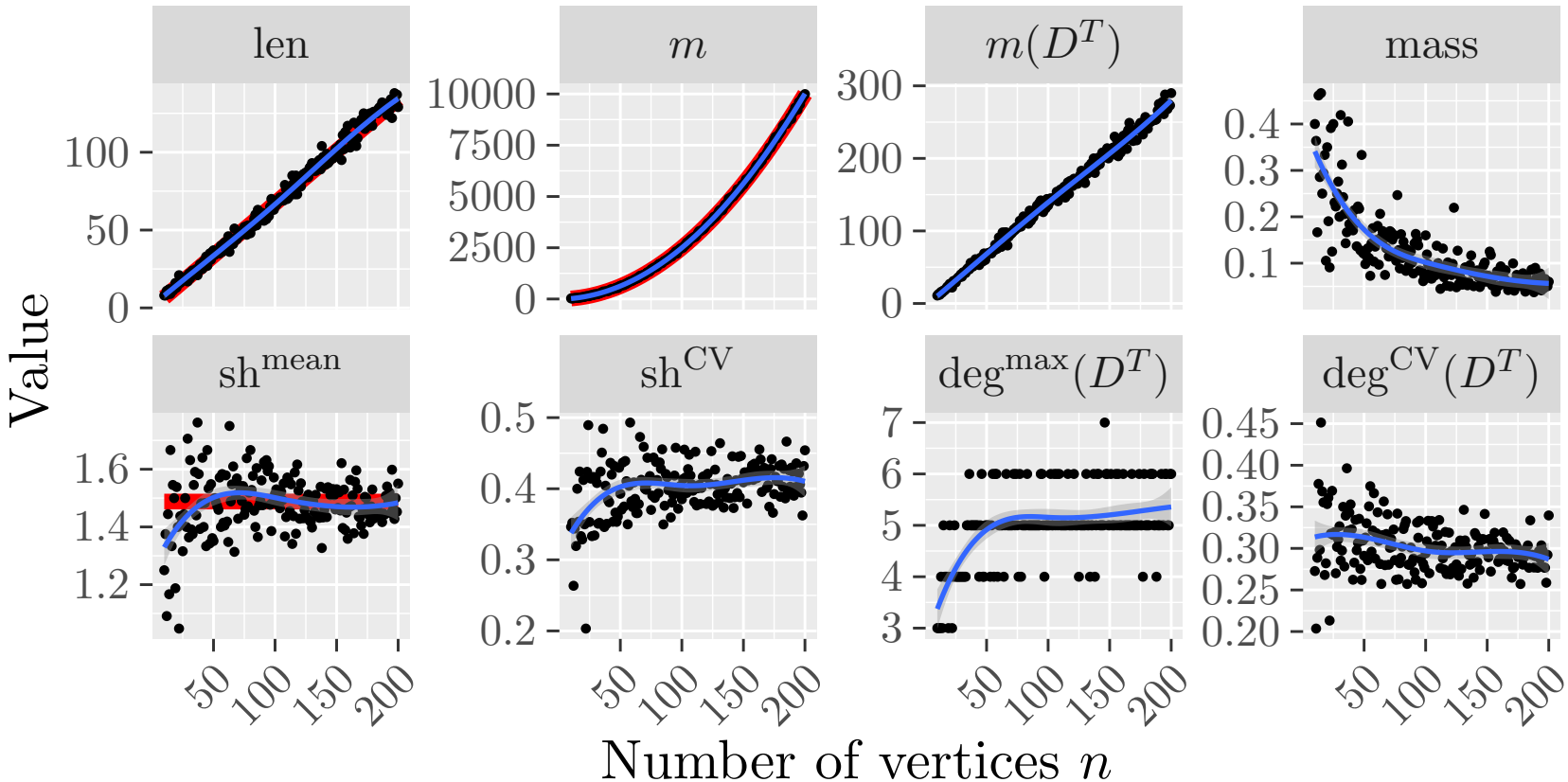
Simple DAGs Properties

DAG	len	m	$m(D^T)$	mass	$\text{shape}^{\text{mean}}$	shape^{CV}	$\text{degree}^{\max}(D^T)$	$\text{degree}^{\text{CV}}$
D_{empty}	1	0	0	1	n	0	0	0
D_{complete}	n	$\frac{n^2}{2}$	n	0	1	0	2	$\frac{1}{\sqrt{2n}}$
D_{chain}	n	n	n	0	1	0	2	$\frac{1}{\sqrt{2n}}$
$D_{\text{out-tree}}$ $D_{\text{in-tree}}$	$\log_2(n)$	n	n	1	$\frac{n}{\log_2(n)}$	$\sqrt{\frac{\log_2(n+1)}{3}}$	3	$\frac{1}{2}$
D_{comb}	$\frac{n}{2}$	n	n	1	2	$\frac{1}{\sqrt{2n}}$	3	$\frac{1}{2}$
D^R_{comb}	$\frac{n}{2}$	n	n	$\frac{1}{2}$	2	$\sqrt{\frac{n}{8}}$	3	$\frac{1}{2}$
$D_{\text{bipartite}}$	2	$\frac{n^2}{4}$	$\frac{n^2}{4}$	1	$\frac{n}{2}$	0	$\frac{n}{2}$	0
D_{square}	$\sqrt{n}$	$n\sqrt{n}$	$n\sqrt{n}$	1	$\sqrt{n}$	0	$2\sqrt{n}$	$\frac{1}{\sqrt{2\sqrt{n}}}$
$D_{\text{triangular}}$	$\sqrt{2n}$	$\frac{2n\sqrt{2n}}{3}$	$\frac{2n\sqrt{2n}}{3}$	1	$\sqrt{\frac{n}{2}}$	$\frac{1}{\sqrt{3}}$	$2\sqrt{2n}$	$\frac{1}{2\sqrt{2}}$

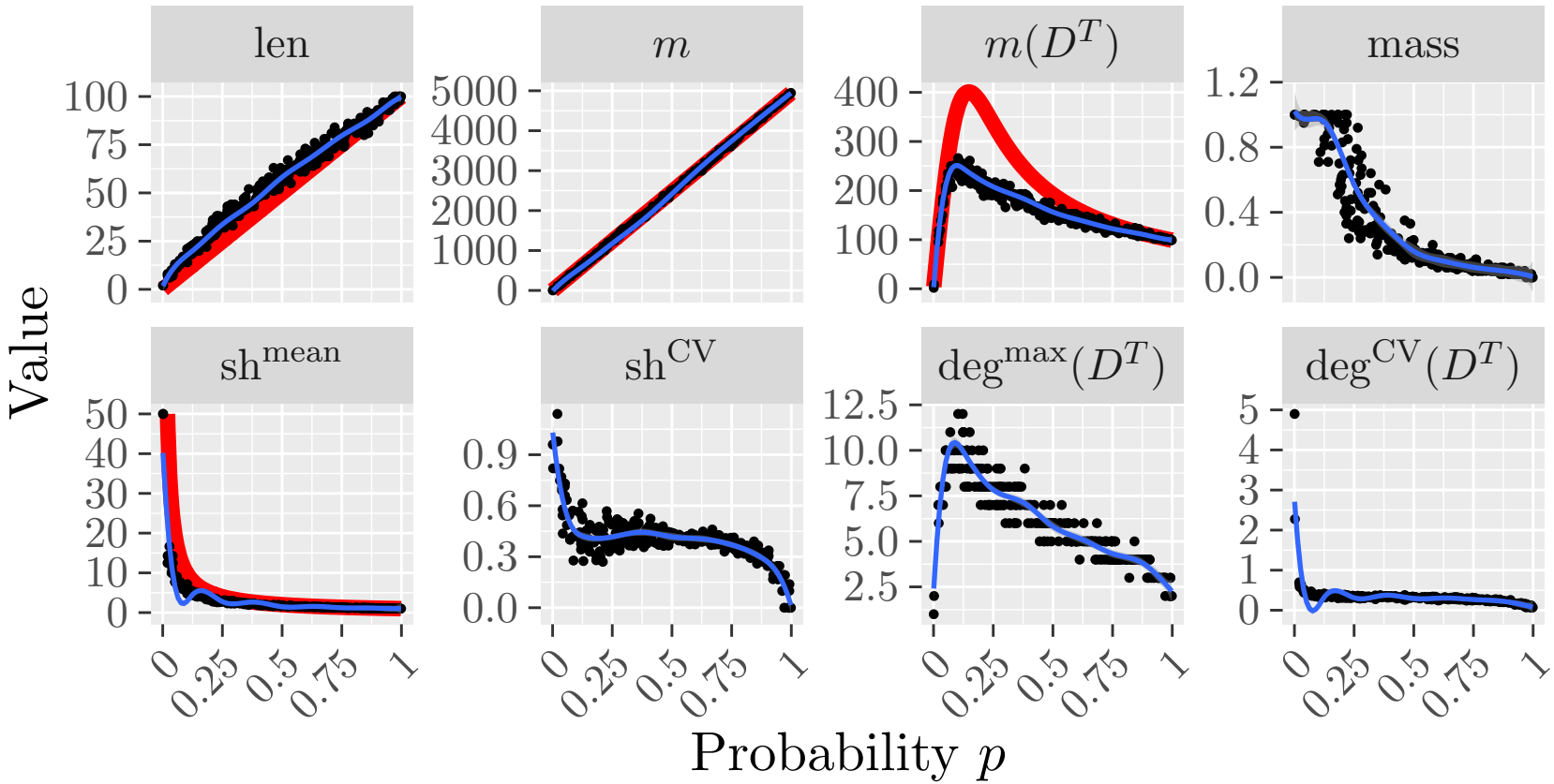
Selected DAGs Properties

length	easy instance when small or large
m	number of edges
$m(D^T)$	number of non-redundant edges
mass	size of largest sub-problem
mean shape	close to the width
shape CV	indicate how the shape varies
max degree	measure local to a single vertex
degree CV	indicate how the degrees vary

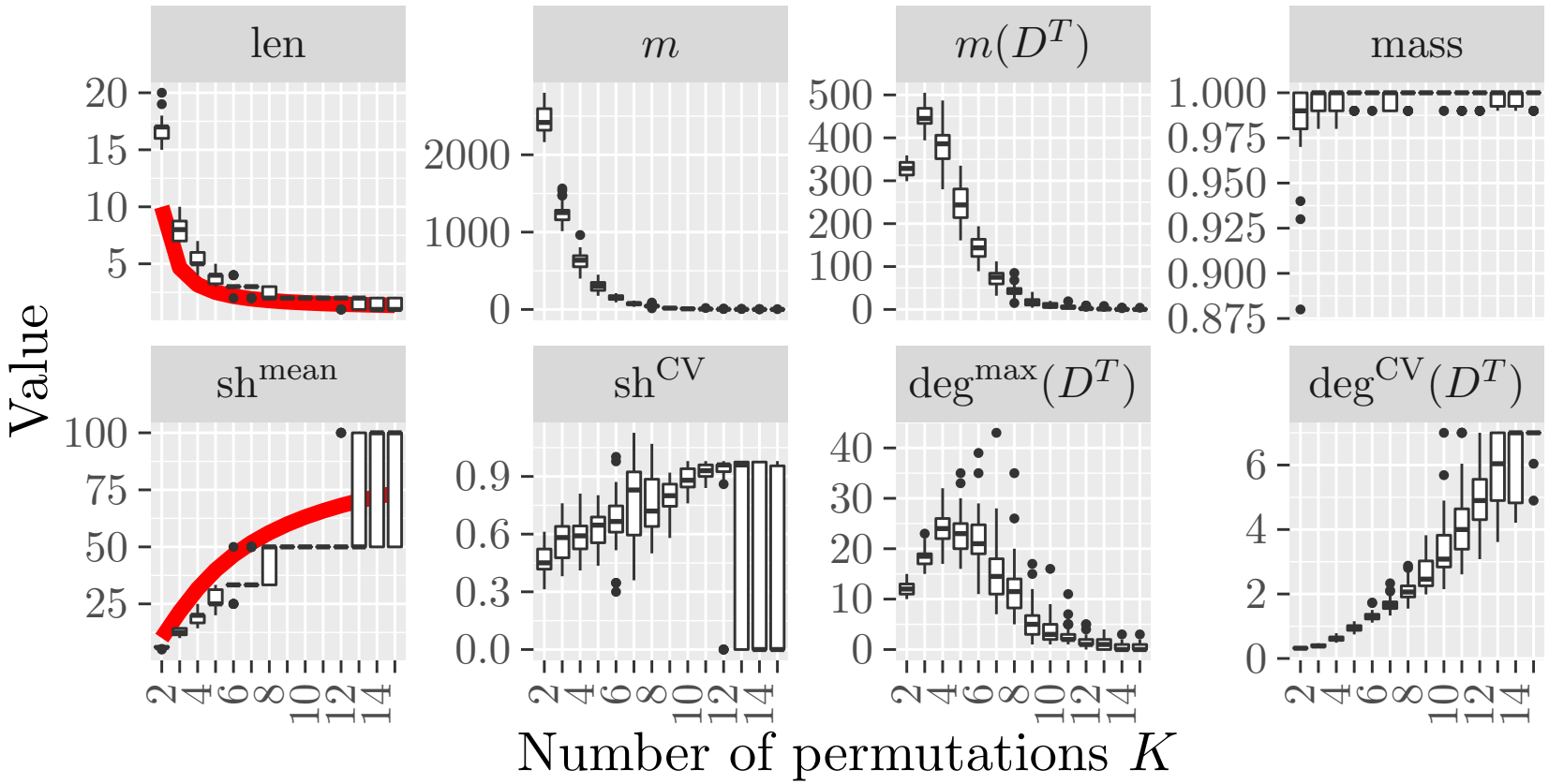
Uniform Analysis



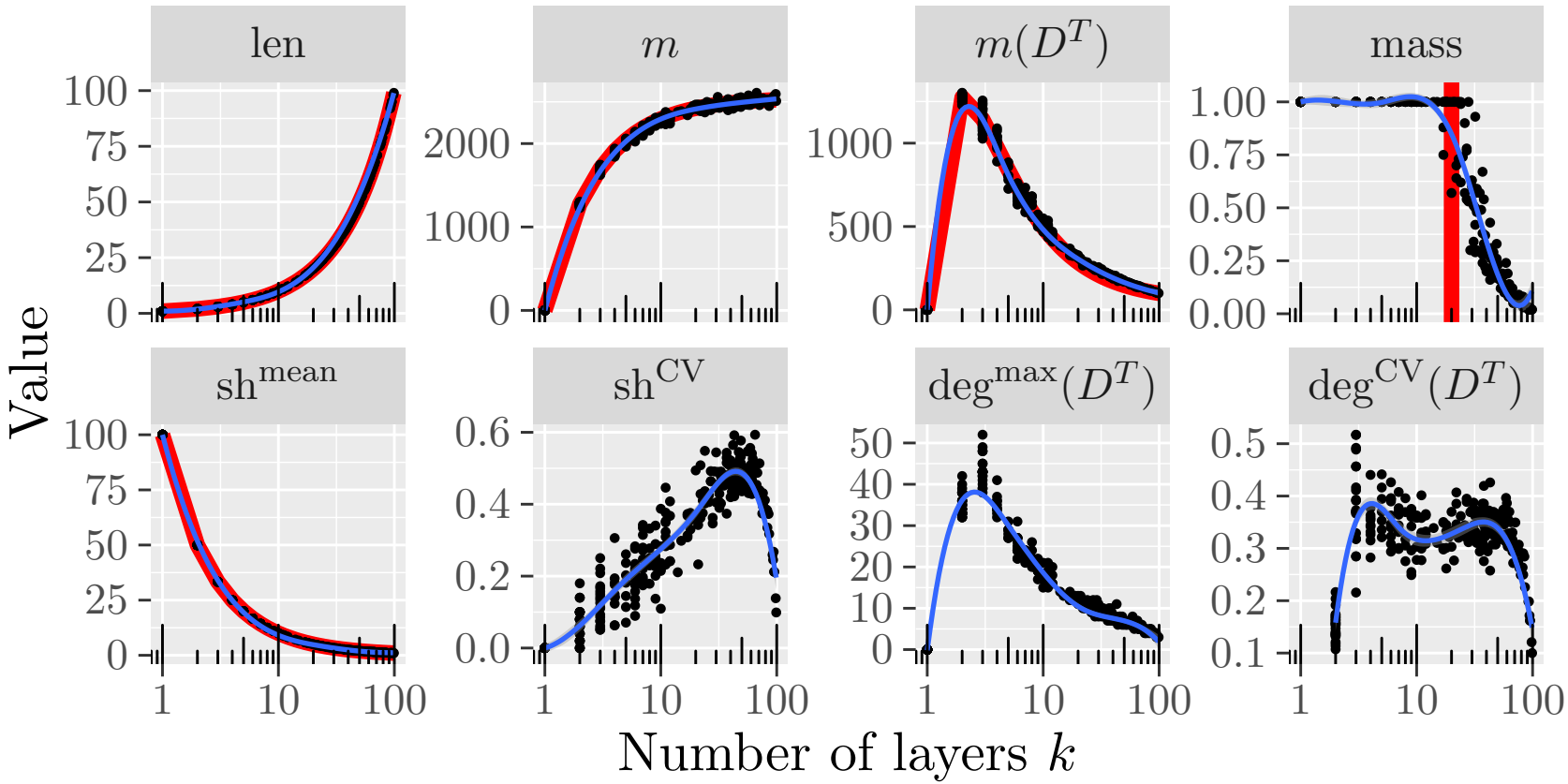
Erdős-Rényi Analysis



Random Order Analysis



Layer-by-Layer Analysis



Conclusion

Summary:

- analyse several DAG properties (e.g. mass)
- provide formal and empirical analysis of existing random generators
- establish the sub-exponential generic time complexity with uniform DAGs

Perspectives:

- generalize properties to non-unitary costs and communications
- design generator uniform on a relevant subspace