Random generation of capacities

Michel GRABISCH[a,b], Christophe LABREUCHE[c]
Peiqi SUN[a]

[a]Université Paris I Panthéon-Sorbonne, Centre d'Economie de la Sorbonne
[b]Paris School of Economics, Paris, France

[c]Thales Research and Technology, Palaiseau, France

Introduction

- Capacities = monotone set functions vanishing on the empty set
  = monotonic TU games

Introduction

- Capacities = monotone set functions vanishing on the empty set
  = monotonic TU games
- Capacities are widely used in decision making and combinatorial problems in OR

Introduction

- Capacities = monotone set functions vanishing on the empty set
  = monotonic TU games
- Capacities are widely used in decision making and combinatorial problems in OR
- Important issue in machine learning: How to randomly generate capacities in a uniform way?

Introduction

- Capacities = monotone set functions vanishing on the empty set
  = monotonic TU games
- Capacities are widely used in decision making and combinatorial problems in OR
- Important issue in machine learning: How to randomly generate capacities in a uniform way?
- Problem theoretically solved but intractable as soon as $n = 5$.

Introduction

- Capacities = monotone set functions vanishing on the empty set
  = monotonic TU games
- Capacities are widely used in decision making and combinatorial problems in OR
- Important issue in machine learning: How to randomly generate capacities in a uniform way?
- Problem theoretically solved but intractable as soon as $n = 5$.
- $\hookrightarrow$ approximation methods are needed: Random Node Generator, Markov chains, two-layer approximation (our proposal), etc.

Introduction

- Capacities = monotone set functions vanishing on the empty set
  = monotonic TU games
- Capacities are widely used in decision making and combinatorial problems in OR
- Important issue in machine learning: How to randomly generate capacities in a uniform way?
- Problem theoretically solved but intractable as soon as $n = 5$.
- $\hookrightarrow$ approximation methods are needed: Random Node Generator, Markov chains, two-layer approximation (our proposal), etc.
- Another problem: How to measure the performance of a capacity generator?

Outline

1. The problem of uniform random generation
2. The 2-layer approximation method
3. Measure of performance

Order polytopes

- $(P, \preccurlyeq)$: *(finite) poset*, with P a finite set and $\preccurlyeq$ a *partial order* (reflexive, antisymmetric, transitive)

Order polytopes

- $(P, \preccurlyeq)$: *(finite) poset*, with P a finite set and $\preccurlyeq$ a *partial order* (reflexive, antisymmetric, transitive)
- $x \in P$ is *maximal* if $x \preccurlyeq y$ with $y \in P$ implies $x = y$. Similarly for *minimal*.

Order polytopes

- $(P, \preccurlyeq)$: *(finite) poset*, with P a finite set and $\preccurlyeq$ a *partial order* (reflexive, antisymmetric, transitive)
- $x \in P$ is *maximal* if $x \preccurlyeq y$ with $y \in P$ implies $x = y$. Similarly for *minimal*.
- $\mathrm{Max}(P, \preccurlyeq)$ and $\mathrm{Min}(P, \preccurlyeq)$ (or simply $\mathrm{Max}(P), \mathrm{Min}(P)$): sets of maximal and minimal elements of P, respectively.

Order polytopes

- $(P, \preccurlyeq)$: *(finite) poset*, with P a finite set and $\preccurlyeq$ a *partial order* (reflexive, antisymmetric, transitive)
- $x \in P$ is *maximal* if $x \preccurlyeq y$ with $y \in P$ implies $x = y$. Similarly for *minimal*.
- $\mathrm{Max}(P, \preccurlyeq)$ and $\mathrm{Min}(P, \preccurlyeq)$ (or simply $\mathrm{Max}(P), \mathrm{Min}(P)$): sets of maximal and minimal elements of P, respectively.
- The *order polytope* (Stanley, 1986) associated to $(P, \preccurlyeq)$, denoted by $\mathcal{O}(P)$, is the set

$$\mathcal{O}(P) = \{f : P \longrightarrow [0, 1] \mid f(x) \leqslant f(y) \text{ if } x \preccurlyeq y\}.$$

Order polytopes

- $(P, \preccurlyeq)$: *(finite) poset*, with P a finite set and $\preccurlyeq$ a *partial order* (reflexive, antisymmetric, transitive)
- $x \in P$ is *maximal* if $x \preccurlyeq y$ with $y \in P$ implies $x = y$. Similarly for *minimal*.
- $\mathrm{Max}(P, \preccurlyeq)$ and $\mathrm{Min}(P, \preccurlyeq)$ (or simply $\mathrm{Max}(P), \mathrm{Min}(P)$): sets of maximal and minimal elements of P, respectively.
- The *order polytope* (Stanley, 1986) associated to $(P, \preccurlyeq)$, denoted by $\mathcal{O}(P)$, is the set

$$\mathcal{O}(P) = \{f : P \longrightarrow [0, 1] \mid f(x) \leqslant f(y) \text{ if } x \preccurlyeq y\}.$$

- $\mathcal{O}(P)$ is a polytope of dimension $p := |P|$.

Linear extensions

- A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$.

Linear extensions

- A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$.
- $E(P)$: set of linear extensions of P; $e(P) = |E(P)|$.

Linear extensions

- A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$.
- $E(P)$: set of linear extensions of P; $e(P) = |E(P)|$.
- Notation: To each linear extension $\leqslant$ on $P = \{x_1, \ldots, x_p\}$ corresponds a permutation σ on $\{1, \ldots, p\}$ such that $x_{\sigma(1)} < \cdots < x_{\sigma(p)}$.

Linear extensions

- A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$.
- $E(P)$: set of linear extensions of P; $e(P) = |E(P)|$.
- Notation: To each linear extension $\leqslant$ on $P = \{x_1, \ldots, x_p\}$ corresponds a permutation σ on $\{1, \ldots, p\}$ such that $x_{\sigma(1)} < \cdots < x_{\sigma(p)}$.
- Each linear extension σ defines a region in $\mathcal{O}(P)$:

$$R_\sigma := \{f \in \mathcal{O}(P) \mid 0 \leqslant f(x_{\sigma(1)}) \leqslant f(x_{\sigma(2)}) \leqslant \cdots \leqslant f(x_{\sigma(p)}) \leqslant 1\}$$

Linear extensions

- A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$.
- $E(P)$: set of linear extensions of P; $e(P) = |E(P)|$.
- Notation: To each linear extension $\leqslant$ on $P = \{x_1, \ldots, x_p\}$ corresponds a permutation σ on $\{1, \ldots, p\}$ such that $x_{\sigma(1)} < \cdots < x_{\sigma(p)}$.
- Each linear extension σ defines a region in $\mathcal{O}(P)$:

$$R_\sigma := \{f \in \mathcal{O}(P) \mid 0 \leqslant f(x_{\sigma(1)}) \leqslant f(x_{\sigma(2)}) \leqslant \cdots \leqslant f(x_{\sigma(p)}) \leqslant 1\}$$

- All regions R_σ are identical (up to a change of coordinates) p-dimensional simplices with volume $\frac{1}{p!}$

Linear extensions

- A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$.
- $E(P)$: set of linear extensions of P; $e(P) = |E(P)|$.
- Notation: To each linear extension $\leqslant$ on $P = \{x_1, \ldots, x_p\}$ corresponds a permutation σ on $\{1, \ldots, p\}$ such that $x_{\sigma(1)} < \cdots < x_{\sigma(p)}$.
- Each linear extension σ defines a region in $\mathcal{O}(P)$:

$$R_\sigma := \{f \in \mathcal{O}(P) \mid 0 \leqslant f(x_{\sigma(1)}) \leqslant f(x_{\sigma(2)}) \leqslant \cdots \leqslant f(x_{\sigma(p)}) \leqslant 1\}$$

- All regions R_σ are identical (up to a change of coordinates) p-dimensional simplices with volume $\frac{1}{p!}$
- Vertices of R_σ are the $p+1$ functions given by

$$0 = f(x_{\sigma(1)}) = \cdots = f(x_{\sigma(k)}),\ f(x_{\sigma(k+1)}) = \cdots = f(x_{\sigma(p)}) = 1,$$

$k = 1, \ldots, p-1$, and the two constant functions 0 and 1.

Random generation of a point in $\mathcal{O}(P)$

- Based on the above facts,

Random generation of a point in $\mathcal{O}(P)$

- Based on the above facts,

  Uniform random generation of a point in $\mathcal{O}(P)$ = uniform random selection of a linear extension σ of P + uniform random selection of a point in R_σ

Random generation of a point in $\mathcal{O}(P)$

- Based on the above facts,

  Uniform random generation of a point in $\mathcal{O}(P)$ = uniform random selection of a linear extension σ of P + uniform random selection of a point in R_σ

- Uniform random selection of a point f in R_σ:

Random generation of a point in $\mathcal{O}(P)$

- Based on the above facts,

  Uniform random generation of a point in $\mathcal{O}(P)$ = uniform random selection of a linear extension σ of P + uniform random selection of a point in R_σ

- Uniform random selection of a point f in R_σ:
  1. Generate p numbers uniformly in $[0, 1]$

Random generation of a point in $\mathcal{O}(P)$

- Based on the above facts,

  Uniform random generation of a point in $\mathcal{O}(P)$ = uniform random selection of a linear extension σ of P + uniform random selection of a point in R_σ

- Uniform random selection of a point f in R_σ:
  1. Generate p numbers uniformly in $[0, 1]$
  2. Order them in increasing order: $z_1 \leqslant \cdots \leqslant z_p$

- Based on the above facts,

  Uniform random generation of a point in $\mathcal{O}(P)$ = uniform random selection of a linear extension σ of P + uniform random selection of a point in R_σ

- Uniform random selection of a point f in R_σ:
  1. Generate p numbers uniformly in $[0, 1]$
  2. Order them in increasing order: $z_1 \leqslant \cdots \leqslant z_p$
  3. Put $f(x_{\sigma(1)}) = z_1, \ldots, f(x_{\sigma(p)}) = z_p$.

Capacities

- $N := \{1, 2, \ldots, n\}$: a finite set of n elements

Capacities

- $N := \{1, 2, \ldots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying

Capacities

- $N := \{1, 2, \ldots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying
  1. $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization)

Capacities

- $N := \{1, 2, \ldots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying
  1. $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization)
  2. $S \subseteq T \Rightarrow \mu(S) \leqslant \mu(T)$ (monotonicity).

- $N := \{1, 2, \dots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying
  1. $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization)
  2. $S \subseteq T \Rightarrow \mu(S) \leqslant \mu(T)$ (monotonicity).
- $\mathcal{C}(N)$: set of capacities on N

Capacities

- $N := \{1, 2, \dots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying
  1. $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization)
  2. $S \subseteq T \Rightarrow \mu(S) \leqslant \mu(T)$ (monotonicity).
- $\mathcal{C}(N)$: set of capacities on N
- $\mathcal{C}(N)$ is an order polytope, whose underlying poset is $(2^N \setminus \{\varnothing, N\}, \subseteq)$

Capacities

- $N := \{1, 2, \dots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying
  1. $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization)
  2. $S \subseteq T \Rightarrow \mu(S) \leqslant \mu(T)$ (monotonicity).
- $\mathcal{C}(N)$: set of capacities on N
- $\mathcal{C}(N)$ is an order polytope, whose underlying poset is $(2^N \setminus \{\varnothing, N\}, \subseteq)$
- Consequence: the problem of the uniform generation of capacities is solved!

Capacities

- $N := \{1, 2, \ldots, n\}$: a finite set of n elements
- A *(normalized) capacity* (Choquet, 1953) on N is a set function $\mu : 2^N \longrightarrow [0, 1]$ satisfying
  1. $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization)
  2. $S \subseteq T \Rightarrow \mu(S) \leqslant \mu(T)$ (monotonicity).
- $\mathcal{C}(N)$: set of capacities on N
- $\mathcal{C}(N)$ is an order polytope, whose underlying poset is $(2^N \setminus \{\varnothing, N\}, \subseteq)$
- Consequence: the problem of the uniform generation of capacities is solved!
- But...

Capacities

The number of linear extensions on $(2^N \setminus \{\varnothing, N\})$ (equal to the number of linear extensions on $(2^N, \subseteq)$ is given in the table below:

Capacities

The number of linear extensions on $(2^N \setminus \{\varnothing, N\})$ (equal to the number of linear extensions on $(2^N, \subseteq)$ is given in the table below:

n	$e(2^N)$
1	1
2	2
3	48
4	1680384
5	14807804035657359360
6	141377911697227887117195970316200795630205476957716480

Capacities

The number of linear extensions on $(2^N \setminus \{\varnothing, N\})$ (equal to the number of linear extensions on $(2^N, \subseteq)$ is given in the table below:

n	$e(2^N)$
1	1
2	2
3	48
4	1680384
5	14807804035657359360
6	141377911697227887117195970316200795630205476957716480

This is sequence A046873 in the *Online Encyclopedia of Integer Sequences*. $e(2^N)$ is not known beyond $n = 7$. Some bounds are known (Brightwell and Winkler, 1991).

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints
- Uniform generation of linear extensions by a Markov chain process (Karzanov and Khachiyan, 1991)(Bubley and Dyer, 1999)

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints
- Uniform generation of linear extensions by a Markov chain process (Karzanov and Khachiyan, 1991)(Bubley and Dyer, 1999)
- Uniform generation of linear extensions by selecting minimal elements with a certain probability (Combarro, Díaz and Miranda, 2013)(Miranda and Garcia-Segador, 2019)(Combarro, Hurtado de Saracho and Díaz, 2019)

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints
- Uniform generation of linear extensions by a Markov chain process (Karzanov and Khachiyan, 1991)(Bubley and Dyer, 1999)
- Uniform generation of linear extensions by selecting minimal elements with a certain probability (Combarro, Díaz and Miranda, 2013)(Miranda and Garcia-Segador, 2019)(Combarro, Hurtado de Saracho and Díaz, 2019)
- Methods specific to some particular families of capacities:

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints
- Uniform generation of linear extensions by a Markov chain process (Karzanov and Khachiyan, 1991)(Bubley and Dyer, 1999)
- Uniform generation of linear extensions by selecting minimal elements with a certain probability (Combarro, Díaz and Miranda, 2013)(Miranda and Garcia-Segador, 2019)(Combarro, Hurtado de Saracho and Díaz, 2019)
- Methods specific to some particular families of capacities:
  - 2-symmetric capacities (Miranda and Garcia-Segador, 2020)

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints
- Uniform generation of linear extensions by a Markov chain process (Karzanov and Khachiyan, 1991)(Bubley and Dyer, 1999)
- Uniform generation of linear extensions by selecting minimal elements with a certain probability (Combarro, Díaz and Miranda, 2013)(Miranda and Garcia-Segador, 2019)(Combarro, Hurtado de Saracho and Díaz, 2019)
- Methods specific to some particular families of capacities:
  - 2-symmetric capacities (Miranda and Garcia-Segador, 2020)
  - supermodular capacities (Beliakov, 2022)

Related literature

- Random Node generator (Havens and Pinar, 2017): select a subset in $2^N \setminus \{\varnothing, N\}$ and draw a number in the interval imposed by monotonicity constraints
- Uniform generation of linear extensions by a Markov chain process (Karzanov and Khachiyan, 1991)(Bubley and Dyer, 1999)
- Uniform generation of linear extensions by selecting minimal elements with a certain probability (Combarro, Díaz and Miranda, 2013)(Miranda and Garcia-Segador, 2019)(Combarro, Hurtado de Saracho and Díaz, 2019)
- Methods specific to some particular families of capacities:
  - 2-symmetric capacities (Miranda and Garcia-Segador, 2020)
  - supermodular capacities (Beliakov, 2022)
  - 2-additive capacities (Miranda and Garcia-Segador, 2020a)

The Random Node Generator

Algorithm:

1. $\mathcal{L} \leftarrow \{\varnothing, N\}$; $\mu(N) = 1$; $\mu(\varnothing) = 0$
2. Pick $S \in 2^N \setminus \mathcal{L}$
3. Compute $\mu_{\min}(S) = \max_{T \in \mathcal{L}, T \subseteq S} \mu(T)$,
   $\mu_{\max}(S) = \min_{T \in \mathcal{L}, T \supseteq S} \mu(T)$
4. Draw uniformly a number t in $[\mu_{\min}(S), \mu_{\max}(S)]$; $\mu(S) \leftarrow t$
5. $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
6. Goto step 2 while $\mathcal{L} \neq 2^N$

The Random Node Generator

Algorithm:

1. $\mathcal{L} \leftarrow \{\varnothing, N\}$; $\mu(N) = 1$; $\mu(\varnothing) = 0$
2. Pick $S \in 2^N \setminus \mathcal{L}$
3. Compute $\mu_{\min}(S) = \max_{T \in \mathcal{L}, T \subseteq S} \mu(T)$,
   $\mu_{\max}(S) = \min_{T \in \mathcal{L}, T \supseteq S} \mu(T)$
4. Draw uniformly a number t in $[\mu_{\min}(S), \mu_{\max}(S)]$; $\mu(S) \leftarrow t$
5. $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
6. Goto step 2 while $\mathcal{L} \neq 2^N$

Advantages: very simple and fast

Drawbacks: yields very biased distribution

The Markov chain generator

- The method generates a linear extension on a poset $(P, \preccurlyeq)$ with a uniform distribution. Let $P = \{x_1, \ldots, x_p\}$.

The Markov chain generator

- The method generates a linear extension on a poset $(P, \preccurlyeq)$ with a uniform distribution. Let $P = \{x_1, \ldots, x_p\}$.
- Two linear extensions σ, τ are *neighbors* if they differ only by a single transposition of neighbor elements:

$$\begin{aligned} \sigma: &\quad x_{\sigma(1)}, \ldots, x_{\sigma(k)}, x_{\sigma(k+1)}, \ldots, x_{\sigma(p)} \\ \tau: &\quad x_{\sigma(1)}, \ldots, x_{\sigma(k+1)}, x_{\sigma(k)}, \ldots, x_{\sigma(p)} \end{aligned}$$

  for some $k \in [1, p-1]$. Denote by $n(\sigma)$ the number of neighbors of σ (at most $p - 1$).

The Markov chain generator

- The method generates a linear extension on a poset $(P, \preccurlyeq)$ with a uniform distribution. Let $P = \{x_1, \dots, x_p\}$.
- Two linear extensions σ, τ are *neighbors* if they differ only by a single transposition of neighbor elements:

$$\sigma : \quad x_{\sigma(1)}, \dots, x_{\sigma(k)}, x_{\sigma(k+1)}, \dots, x_{\sigma(p)}$$

$$\tau : \quad x_{\sigma(1)}, \dots, x_{\sigma(k+1)}, x_{\sigma(k)}, \dots, x_{\sigma(p)}$$

  for some $k \in [1, p-1]$. Denote by $n(\sigma)$ the number of neighbors of σ (at most $p-1$).
- The *order Markov chain M* is defined on the set of states $E(P)$ with transition probabilities:

$$P(\sigma, \tau) = \begin{cases} 1/(2p-2) & \text{if } \sigma, \tau \text{ are neighbors} \\ 1 - n(\sigma)/(2p-2) & \text{if } \sigma = \tau \\ 0 & \text{otherwise.} \end{cases}$$

The Markov chain generator

- The method generates a linear extension on a poset $(P, \preccurlyeq)$ with a uniform distribution. Let $P = \{x_1, \dots, x_p\}$.
- Two linear extensions σ, τ are *neighbors* if they differ only by a single transposition of neighbor elements:

$$\begin{aligned} \sigma : &\quad x_{\sigma(1)}, \dots, x_{\sigma(k)}, x_{\sigma(k+1)}, \dots, x_{\sigma(p)} \\ \tau : &\quad x_{\sigma(1)}, \dots, x_{\sigma(k+1)}, x_{\sigma(k)}, \dots, x_{\sigma(p)} \end{aligned}$$

  for some $k \in [1, p-1]$. Denote by $n(\sigma)$ the number of neighbors of σ (at most $p-1$).
- The *order Markov chain M* is defined on the set of states $E(P)$ with transition probabilities:

$$P(\sigma, \tau) = \begin{cases} 1/(2p-2) & \text{if } \sigma, \tau \text{ are neighbors} \\ 1 - n(\sigma)/(2p-2) & \text{if } \sigma = \tau \\ 0 & \text{otherwise.} \end{cases}$$

- The order Markov chain M is ergodic time-reversible and converges to the uniform distribution on $E(P)$.

Algorithm

1. Input: a poset $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$, an integer T
2. Find a linear extension σ on P
3. For $i = 1$ to T do:
   - choose at random an integer $k \in [1, 2p - 2]$
   - if $k \leqslant p - 1$ and not$[x_{\sigma(k)} \prec x_{\sigma(k+1)}]$ then swap $x_{\sigma(k)}$ and $x_{\sigma(k+1)}$ in σ
4. Output σ

The Markov chain generator

Algorithm

1. Input: a poset $(P, \preccurlyeq)$ with $P = \{x_1, \dots, x_p\}$, an integer T
2. Find a linear extension σ on P
3. For $i = 1$ to T do:
   - choose at random an integer $k \in [1, 2p-2]$
   - if $k \leqslant p-1$ and not$[x_{\sigma(k)} \prec x_{\sigma(k+1)}]$ then swap $x_{\sigma(k)}$ and $x_{\sigma(k+1)}$ in σ
4. Output σ

Estimation of T to get almost uniformity: $T = O(p^5 \log(e(P)))$.

Outline

1. The problem of uniform random generation
2. **The 2-layer approximation method**
3. Measure of performance

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$.

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \dots, x_p\}$.
- Any linear extension $x_{\sigma(1)}, \dots, x_{\sigma(p)}$ satisfies $x_{\sigma(1)} \in \mathrm{Min}(P)$ and $x_{\sigma(p)} \in \mathrm{Max}(P)$.

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$.
- Any linear extension $x_{\sigma(1)}, \ldots, x_{\sigma(p)}$ satisfies $x_{\sigma(1)} \in \mathrm{Min}(P)$ and $x_{\sigma(p)} \in \mathrm{Max}(P)$.
- Similarly, $x_{\sigma(2)} \in \mathrm{Min}(P \setminus \{x_{\sigma(1)}\})$, and $x_{\sigma(p-1)} \in \mathrm{Max}(P \setminus \{x_{\sigma(p)}\})$.

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \dots, x_p\}$.
- Any linear extension $x_{\sigma(1)}, \dots, x_{\sigma(p)}$ satisfies $x_{\sigma(1)} \in \mathrm{Min}(P)$ and $x_{\sigma(p)} \in \mathrm{Max}(P)$.
- Similarly, $x_{\sigma(2)} \in \mathrm{Min}(P \setminus \{x_{\sigma(1)}\})$, and $x_{\sigma(p-1)} \in \mathrm{Max}(P \setminus \{x_{\sigma(p)}\})$.
- Based on this observation, the probability that a linear extension of P starts with $m \in \mathrm{Min}(P)$ (resp., ends with $M \in \mathrm{Max}(P)$) is

$$\Pr(m \mid P) = \frac{e(P \setminus \{m\})}{e(P)}; \quad \Pr(M \mid P) = \frac{e(P \setminus \{M\})}{e(P)}$$

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$.
- Any linear extension $x_{\sigma(1)}, \ldots, x_{\sigma(p)}$ satisfies $x_{\sigma(1)} \in \mathrm{Min}(P)$ and $x_{\sigma(p)} \in \mathrm{Max}(P)$.
- Similarly, $x_{\sigma(2)} \in \mathrm{Min}(P \setminus \{x_{\sigma(1)}\})$, and $x_{\sigma(p-1)} \in \mathrm{Max}(P \setminus \{x_{\sigma(p)}\})$.
- Based on this observation, the probability that a linear extension of P starts with $m \in \mathrm{Min}(P)$ (resp., ends with $M \in \mathrm{Max}(P)$) is

$$\Pr(m \mid P) = \frac{e(P \setminus \{m\})}{e(P)}; \quad \Pr(M \mid P) = \frac{e(P \setminus \{M\})}{e(P)}$$

- $\hookrightarrow$ generating a linear extension amounts to choosing, according to the correct probability given above, a minimal or a maximal element of a poset which is diminished by one element at each step.

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \dots, x_p\}$.
- Any linear extension $x_{\sigma(1)}, \dots, x_{\sigma(p)}$ satisfies $x_{\sigma(1)} \in \mathrm{Min}(P)$ and $x_{\sigma(p)} \in \mathrm{Max}(P)$.
- Similarly, $x_{\sigma(2)} \in \mathrm{Min}(P \setminus \{x_{\sigma(1)}\})$, and $x_{\sigma(p-1)} \in \mathrm{Max}(P \setminus \{x_{\sigma(p)}\})$.
- Based on this observation, the probability that a linear extension of P starts with $m \in \mathrm{Min}(P)$ (resp., ends with $M \in \mathrm{Max}(P)$) is

$$\Pr(m \mid P) = \frac{e(P \setminus \{m\})}{e(P)}; \quad \Pr(M \mid P) = \frac{e(P \setminus \{M\})}{e(P)}$$

- $\hookrightarrow$ generating a linear extension amounts to choosing, according to the correct probability given above, a minimal or a maximal element of a poset which is diminished by one element at each step.
- As this probability directly depends on $e(P)$, the computation can be exact only when P becomes small enough.

Basic idea

- Let $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$.
- Any linear extension $x_{\sigma(1)}, \ldots, x_{\sigma(p)}$ satisfies $x_{\sigma(1)} \in \mathrm{Min}(P)$ and $x_{\sigma(p)} \in \mathrm{Max}(P)$.
- Similarly, $x_{\sigma(2)} \in \mathrm{Min}(P \setminus \{x_{\sigma(1)}\})$, and $x_{\sigma(p-1)} \in \mathrm{Max}(P \setminus \{x_{\sigma(p)}\})$.
- Based on this observation, the probability that a linear extension of P starts with $m \in \mathrm{Min}(P)$ (resp., ends with $M \in \mathrm{Max}(P)$) is

$$\Pr(m \mid P) = \frac{e(P \setminus \{m\})}{e(P)}; \quad \Pr(M \mid P) = \frac{e(P \setminus \{M\})}{e(P)}$$

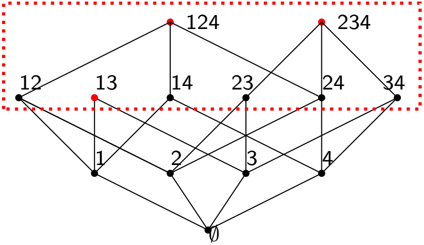
- $\hookrightarrow$ generating a linear extension amounts to choosing, according to the correct probability given above, a minimal or a maximal element of a poset which is diminished by one element at each step.
- As this probability directly depends on $e(P)$, the computation can be exact only when P becomes small enough.
- Idea: take the lower part of the poset for choosing minimal elements, and the upper part for choosing maximal elements, thus neglecting minimal and maximal elements which are outside these two subparts.

The approximation method

- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.

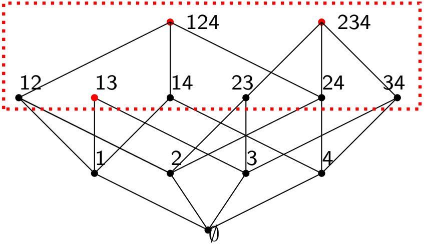
- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.
- *Layer* of H: all nodes (subsets) of same cardinality

- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.
- *Layer* of H: all nodes (subsets) of same cardinality
- T_H: the two top layers of H. When needed, we specify $T_H[h, k, |I|]$, where



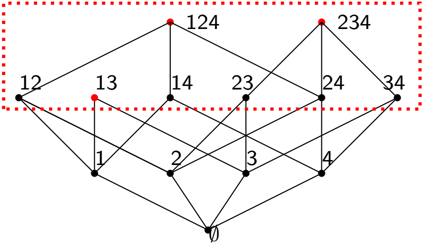
The approximation method

- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.
- *Layer* of H: all nodes (subsets) of same cardinality
- T_H: the two top layers of H. When needed, we specify $T_H[h, k, |I|]$, where
  - h: number of nodes in the upper layer of T_H (2)



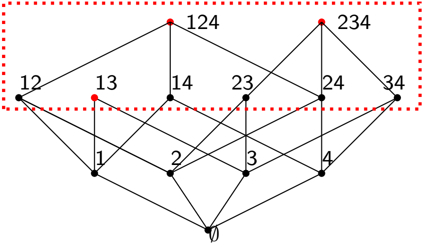
The approximation method

- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.
- *Layer* of H: all nodes (subsets) of same cardinality
- T_H: the two top layers of H. When needed, we specify $T_H[h, k, |I|]$, where
  - h: number of nodes in the upper layer of T_H (2)
  - k: number of nodes in the lower layer of T_H (6)



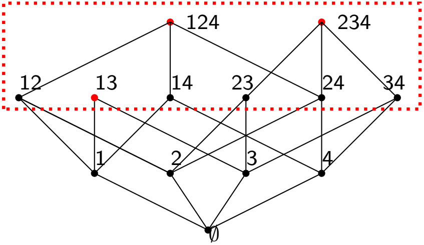
The approximation method

- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.
- *Layer* of H: all nodes (subsets) of same cardinality
- T_H: the two top layers of H. When needed, we specify $T_H[h, k, |I|]$, where
  - h: number of nodes in the upper layer of T_H (2)
  - k: number of nodes in the lower layer of T_H (6)
  - I: set of isolated nodes in the 2nd layer ($\{13\}$)



The approximation method

- We take $P = 2^N \setminus \{\varnothing, N\}$ and delete step by step minimal and maximal elements. Generically, we call $(H, \subseteq)$ the current poset.
- *Layer* of H: all nodes (subsets) of same cardinality
- T_H: the two top layers of H. When needed, we specify $T_H[h, k, |I|]$, where
  - h: number of nodes in the upper layer of T_H (2)
  - k: number of nodes in the lower layer of T_H (6)
  - I: set of isolated nodes in the 2nd layer ($\{13\}$)



- For y in the 2nd layer, pred(y) is the set of its predecessors in the 1st layer (similarly with succ(x), x in the 1st layer)

The approximation method

- In a dual way, we introduce B_H (denoted also $B_H[h', k', |I'|]$), the poset of the two bottom layers of H.

The approximation method

- In a dual way, we introduce B_H (denoted also $B_H[h', k', |I'|]$), the poset of the two bottom layers of H.
- Consider a maximal element M of H belonging to T_H, and a minimal element m of H belonging to B_H. We put

$$\Pr(M \mid H) \approx \frac{e(T_H \setminus \{M\})}{e(T_H)} = \Pr(M \mid T_H)$$

$$\Pr(m \mid H) \approx \frac{e(B_H \setminus \{m\})}{e(B_H)} = \Pr(m \mid B_H)$$

The approximation method

- In a dual way, we introduce B_H (denoted also $B_H[h', k', |I'|]$), the poset of the two bottom layers of H.
- Consider a maximal element M of H belonging to T_H, and a minimal element m of H belonging to B_H. We put

$$\mathrm{Pr}(M \mid H) \approx \frac{e(T_H \setminus \{M\})}{e(T_H)} = \mathrm{Pr}(M \mid T_H)$$
$$\mathrm{Pr}(m \mid H) \approx \frac{e(B_H \setminus \{m\})}{e(B_H)} = \mathrm{Pr}(m \mid B_H)$$

- Rationale: If in average a node has ℓ predecessors in the layer just above, a node in layer k has therefore $O(\ell^{k-1})$ predecessors in H. Hence, a node in the 3d layer has very little probability to become maximal, since all its predecessors must be eliminated first, *without eliminating all nodes of the 1st layer*

The algorithm

generate-linext(P, l)
Input: a poset P subset of $2^N \setminus \{\varnothing, N\}$
Output: a linear extension l of P generated with a uniform distribution
$H \leftarrow P$; $lmin \leftarrow \varnothing$; $lmax \leftarrow \varnothing$
While height of $H > 2$ **do**

Compute the basic parameters of T_H: $k, h, |I|$
Select $M \in T_H[h, k, |I|]$ with probability $\Pr(M \mid T_H[h, k, |I|])$
Add M at the beginning of $lmax$
Compute the basic parameters of B_H: $k', h', |I'|$
Select $m \in B_H[h', k', |I'|]$ with probability $\Pr(m \mid B_H[h', k', |I'|])$
Add m at the end of $lmin$

$H \leftarrow H \setminus \{M, m\}$
end while

% Now H is reduced to two layers: B_H and T_H coincide

While height of $H = 2$ **do**

If number of nodes in the upper layer $\leqslant$ number of nodes in the lower layer **then**

Select $M \in T_H[h, k, |I|]$ with probability $\mathrm{Pr}(M \mid T_H[h, k, |I|])$

Add M at the beginning of *lmax*

$H \leftarrow H \setminus \{M\}$

otherwise

Select $m \in B_H[h', k', |I'|]$ with probability $\mathrm{Pr}(m \mid B_H[h', k', |I'|])$

Add m at the end of *lmin*

$H \leftarrow H \setminus \{m\}$

end if

end while

% Now H is reduced to one layer, which is an antichain whose elements have
% the same probability

While $H \neq \varnothing$ **do**

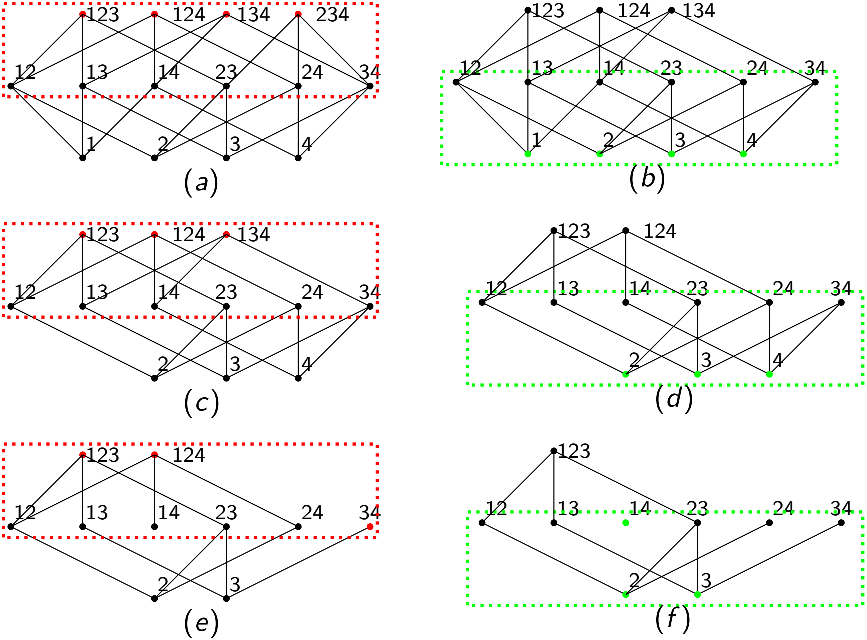
Select uniformly at random an element $x \in H$

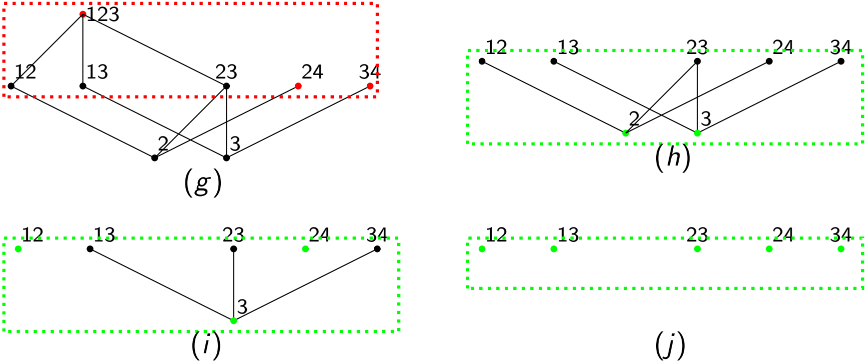
Add x at the end of *lmin*

$H \leftarrow H \setminus \{x\}$

end while

$l \leftarrow lmin$; concatenate *lmax* to the end of l


123 124 134 234
12 13 14 23 24 34
1 2 3 4
(a)
123 124 134
12 13 14 23 24 34
1 2 3 4
(b)
123 124 134
12 13 14 23 24 34
2 3 4
(c)
123 124
12 13 14 23 24 34
2 3 4
(d)
123 124
12 13 14 23 24 34
2 3
(e)
123
12 13 14 23 24 34
2 3
(f)



Resulting linear extension: 1, 4, 14, 2, 3, 12, 34, 24, 23, 13, 123, 124, 134, 234

Computation of the probabilities

It remains to compute $\Pr(M \mid T_H)$ and $\Pr(m \mid B_H)$. This is made possible through a simplifying assumption.

Computation of the probabilities

It remains to compute $\Pr(M \mid T_H)$ and $\Pr(m \mid B_H)$. This is made possible through a simplifying assumption.

Definition

Let x be a node of the upper layer of T_H.

1. The function f_x assigns to every node y of the lower layer an integer as follows:

$$f_x(y) = \begin{cases} |\text{pred}(y)|, & y \in \text{succ}(x) \\ 0, & \text{otherwise.} \end{cases}$$

2. The function $n_x : \mathbb{N} \to \mathbb{N}$ is defined from f_x as follows: $n_x(r)$ is the number of occurences of $f_x(y) = r$, i.e., $n_x(r) = |f_x^{-1}(r)|$. When $r > 0$, it is the number of successors of x having r predecessors, otherwise when $r = 0$ it is the number of nodes in the lower layer which are not successors of x.

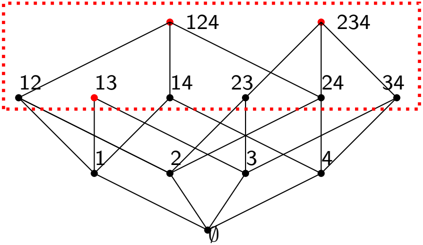
Computation of the probabilities

Definition

We say that T_H is *regular* if n_x is invariant with x, i.e., $n_x(r) = n_{x'}(r)$ for every r and every two nodes x, x' in the upper layer.

Definition

We say that T_H is *regular* if n_x is invariant with x, i.e., $n_x(r) = n_{x'}(r)$ for every r and every two nodes x, x' in the upper layer.



$n_{124}(0) = 3 = n_{234}(0)$, $n_{124}(1) = 2 = n_{234}(1)$, $n_{124}(2) = 1 = n_{234}(2)$, hence the above T_H is regular. Every T_H closed under intersection and balanced is regular.

Dual definitions exist for B_H.

Computation of the probabilities

Proposition

Consider the poset $T_H[h,k,|I|]$ and suppose that it is regular. Then the probabilities $\mathbb{P}_u(T_H[h,k,|I|])$ that node x of the upper layer terminates a linear extension, and $\mathbb{P}_l(T_H[h,k,|I|])$ that isolated node y of the lower layer terminates a linear extension are given by

$$\mathbb{P}_u(T_H[h,k,|I|]) = \frac{1}{h}\frac{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i)}{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i) + |I| \times \prod_{i=1}^{|I''|}(h-1+k-|I'|+i)\prod_{i=1}^{|I|-1}(h+k-|I|+i)}$$

$$\mathbb{P}_l(T_H[h,k,|I|]) = \frac{\prod_{i=1}^{|I''|}(h-1+k-|I'|+i)\prod_{i=1}^{|I|-1}(h+k-|I|+i)}{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i) + |I| \times \prod_{i=1}^{|I''|}(h-1+k-|I'|+i)\prod_{i=1}^{|I|-1}(h+k-|I|+i)},$$

where I' is the set of isolated nodes in the poset $T_{H\setminus\{x\}}$, and $I \cup I'' = I'$.

Outline

Distribution of $\mu(S)$

- Uniform distribution of μ in $\mathcal{C}(N)$ does not mean that the distribution of $\mu(S)$ for a given S is uniform!

Distribution of $\mu(S)$

- Uniform distribution of μ in $\mathcal{C}(N)$ does not mean that the distribution of $\mu(S)$ for a given S is uniform!
- Exact distribution of $\mu(S)$ seems to be very hard to obtain.

Distribution of $\mu(S)$

- Uniform distribution of μ in $\mathcal{C}(N)$ does not mean that the distribution of $\mu(S)$ for a given S is uniform!
- Exact distribution of $\mu(S)$ seems to be very hard to obtain.
- Denote by $\boldsymbol{\mu}$ the r.v. with uniform distribution on $\mathcal{C}(N)$. Take a linear extension σ and consider the associated region $R_\sigma \subseteq \mathcal{C}(N)$.

Distribution of $\mu(S)$

- Uniform distribution of μ in $\mathcal{C}(N)$ does not mean that the distribution of $\mu(S)$ for a given S is uniform!
- Exact distribution of $\mu(S)$ seems to be very hard to obtain.
- Denote by $\boldsymbol{\mu}$ the r.v. with uniform distribution on $\mathcal{C}(N)$. Take a linear extension σ and consider the associated region $R_\sigma \subseteq \mathcal{C}(N)$.
- Given that $\boldsymbol{\mu} \in R_\sigma$, we know that $\boldsymbol{\mu}(S_{\sigma(k)})$ follows the distribution of the kth order statistics on $[0,1]$. It is known that the probability density function $f_{(k)}$ of the kth order statistics on $[0,1]$ when the underlying $2^n - 2$ random variables are i.i.d. and uniform is a Beta distribution:

$$f_{(k)}(u) = (2^n - 2)\binom{2^n - 3}{k - 1}(1-u)^{2^n-2-k}u^{k-1} = \mathrm{Beta}(k, 2^n + k - 1)$$

Distribution of $\mu(S)$

Denoting by OS_k the corresponding cumulative distribution function, it follows that for any $S \in 2^N \setminus \{\varnothing, N\}$, the distribution $F_{\boldsymbol{\mu}(S)}(\alpha)$ is given by

$$\begin{aligned}
F_{\boldsymbol{\mu}(S)}(\alpha) = \mathrm{Pr}(\boldsymbol{\mu}(S) \leqslant \alpha) &= \sum_{\sigma \in E(2^N \setminus \{\varnothing, N\})} \mathrm{Pr}(\boldsymbol{\mu}(S) \leqslant \alpha \mid \boldsymbol{\mu} \in R_\sigma)\mathrm{Pr}(\boldsymbol{\mu} \in R_\sigma) \\
&= \frac{1}{e(2^N)} \sum_{\sigma \in E(2^N \setminus \{\varnothing, N\})} \mathrm{Pr}(\boldsymbol{\mu}(S) \leqslant \alpha \mid \boldsymbol{\mu} \in R_\sigma) \\
&= \frac{1}{e(2^N)} \sum_{\sigma \in E(2^N \setminus \{\varnothing, N\})} \mathrm{OS}_{k(S,\sigma)}(\alpha), \qquad (1)
\end{aligned}$$

where $E(2^N \setminus \{\varnothing, N\})$ is the set of permutations corresponding to linear extensions, and $k(S, \sigma)$ is such that $S = S_{\sigma(k)}$.

Distribution of $\mu(S)$

Lemma

Assume $\boldsymbol{\mu}$ *is uniformly distributed and take* $\varnothing \neq S, S' \subset N$. *Then*

1. $\boldsymbol{\mu}(S)$ *and* $\boldsymbol{\mu}(S')$ *for* $|S| = |S'|$ *are identically distributed.*
2. $\boldsymbol{\mu}(S)$ *and* $1 - \boldsymbol{\mu}(N \setminus S)$ *are identically distributed.*

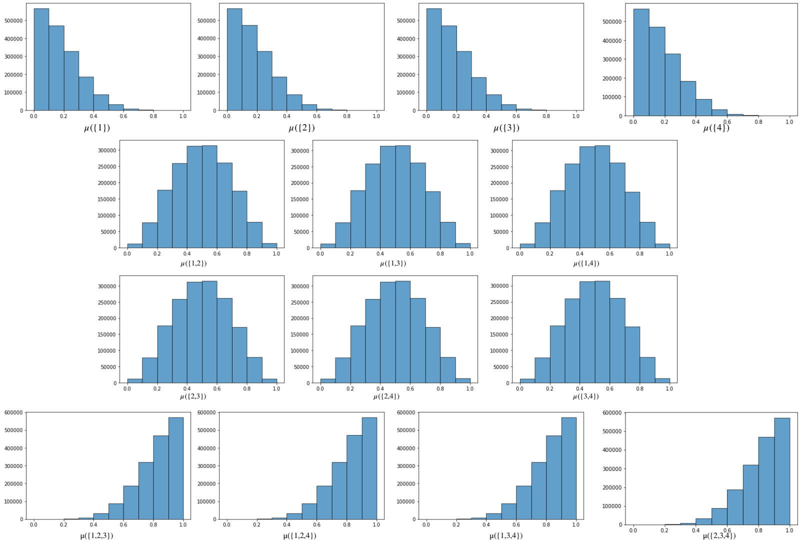


Figure: Histograms of $\mu(S)$ for $n = 4$ and the exact method

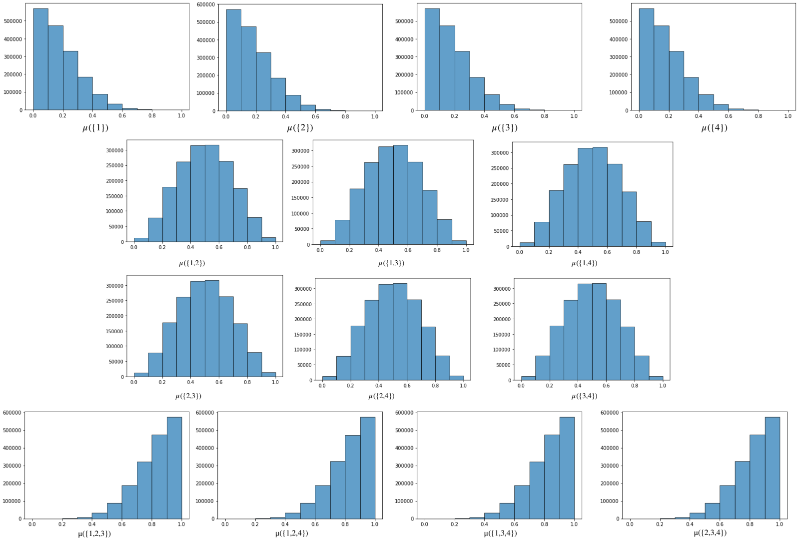


Figure: Histograms of $\mu(S)$ for $n = 4$ and the 2-layer approximation method

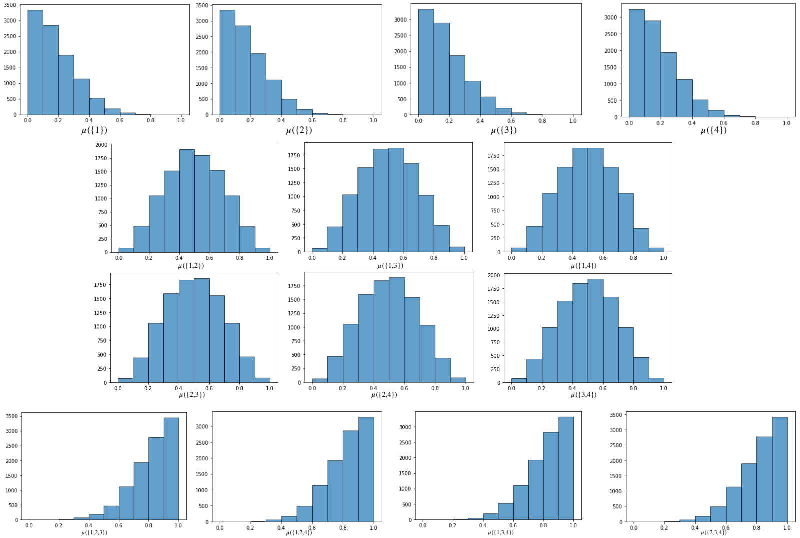


Figure: Histograms of $\mu(S)$ for $n = 4$ and the Markov chain method

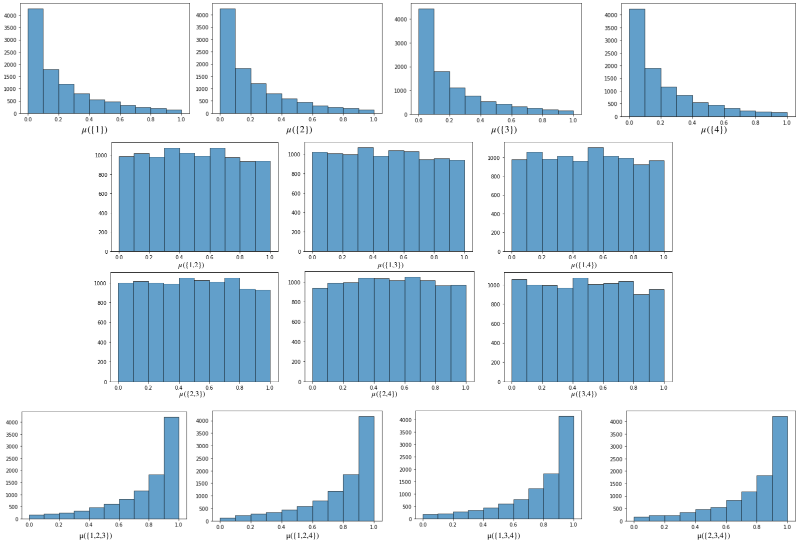


Figure: Histograms of $\mu(S)$ for $n = 4$ and the Random Node Generator

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$
- centroid of $\mathcal{C}(N) \neq$ barycenter of $\mathcal{C}(N)$ (=average of vertices)!

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$
- centroid of $\mathcal{C}(N) \neq$ barycenter of $\mathcal{C}(N)$ (=average of vertices)!
- We must use the triangulation of $\mathcal{C}(N)$ into the regions R_σ in order to compute the centroid c:

$$c = \sum_{\sigma \in E(2^N)} b_\sigma$$

with b_σ the barycenter of R_σ

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$
- centroid of $\mathcal{C}(N) \neq$ barycenter of $\mathcal{C}(N)$ (=average of vertices)!
- We must use the triangulation of $\mathcal{C}(N)$ into the regions R_σ in order to compute the centroid c:

$$c = \sum_{\sigma \in E(2^N)} b_\sigma$$

with b_σ the barycenter of R_σ

- Consequently, the exact centroid c can be computed for $n \leqslant 4$ only.

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$
- centroid of $\mathcal{C}(N) \neq$ barycenter of $\mathcal{C}(N)$ (=average of vertices)!
- We must use the triangulation of $\mathcal{C}(N)$ into the regions R_σ in order to compute the centroid c:

$$c = \sum_{\sigma \in E(2^N)} b_\sigma$$

  with b_σ the barycenter of R_σ
- Consequently, the exact centroid c can be computed for $n \leqslant 4$ only.
- The centroid inherits the properties of $\mu(S)$, i.e., $c(N \setminus S) = 1 - c(S)$ and $c(S)$ depends on $|S|$ only.

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$
- centroid of $\mathcal{C}(N) \neq$ barycenter of $\mathcal{C}(N)$ (=average of vertices)!
- We must use the triangulation of $\mathcal{C}(N)$ into the regions R_σ in order to compute the centroid c:

$$c = \sum_{\sigma \in E(2^N)} b_\sigma$$

  with b_σ the barycenter of R_σ
- Consequently, the exact centroid c can be computed for $n \leqslant 4$ only.
- The centroid inherits the properties of $\mu(S)$, i.e., $c(N \setminus S) = 1 - c(S)$ and $c(S)$ depends on $|S|$ only.
- centroid for $n = 3$:

$$c = (0.298, 0.298, 0.298, 0.702, 0.702, 0.702).$$

The centroid of $\mathcal{C}(N)$

- Idea: checking if the average over generated capacities = centroid c of $\mathcal{C}(N)$ is a good measure of the homogeneity of repartition in $\mathcal{C}(N)$
- centroid of $\mathcal{C}(N) \neq$ barycenter of $\mathcal{C}(N)$ (=average of vertices)!
- We must use the triangulation of $\mathcal{C}(N)$ into the regions R_σ in order to compute the centroid c:

$$c = \sum_{\sigma \in E(2^N)} b_\sigma$$

  with b_σ the barycenter of R_σ
- Consequently, the exact centroid c can be computed for $n \leqslant 4$ only.
- The centroid inherits the properties of $\mu(S)$, i.e., $c(N \setminus S) = 1 - c(S)$ and $c(S)$ depends on $|S|$ only.
- centroid for $n = 3$:

$$c = (0.298, 0.298, 0.298, 0.702, 0.702, 0.702).$$

- centroid for $n = 4$:

$$c = (0.1810, 0.1810, 0.1810, 0.1810, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.8190, 0.8190, 0.8190, 0.8190).$$

Experimental results

- Idea: compare with the Markov chain method (asymptotically exact when T tends to infinity), by choosing T so that the performance of the 2-layer approximation method is approximately the same as the one of the Markov chain method

Experimental results

- Idea: compare with the Markov chain method (asymptotically exact when T tends to infinity), by choosing T so that the performance of the 2-layer approximation method is approximately the same as the one of the Markov chain method
- When $n \leqslant 4$, we take as performance the L_1 distance between the theoretical centroid and the obtained centroid. We obtain $T = 1170$.

Experimental results

- Idea: compare with the Markov chain method (asymptotically exact when T tends to infinity), by choosing T so that the performance of the 2-layer approximation method is approximately the same as the one of the Markov chain method
- When $n \leqslant 4$, we take as performance the L_1 distance between the theoretical centroid and the obtained centroid. We obtain $T = 1170$.
- When $n > 4$, we use the symmetry properties of the centroid ($c(S)$ depends only on $|S|$). The performance is measured by the standard deviation of $c(S)$ when $|S|$ is constant. We obtain $T = 9000$ for $n = 5$.

Experimental results: Comparison of distributions

- A quantitative comparison is done by the Kullback-Leibler divergence.

Experimental results: Comparison of distributions

- A quantitative comparison is done by the Kullback-Leibler divergence.
- Given two discrete probability distributions p, q on the same universe X, the *Kullback-Leibler divergence* is defined as

$$\mathbb{D}_{KL}(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

The smaller the value, the closer are the two distributions.

Experimental results: Comparison of distributions

- A quantitative comparison is done by the Kullback-Leibler divergence.
- Given two discrete probability distributions p, q on the same universe X, the *Kullback-Leibler divergence* is defined as

$$\mathbb{D}_{KL}(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

  The smaller the value, the closer are the two distributions.
- Distributions of $\boldsymbol{\mu}(S)$ are discretized with $\delta = 0.01$ on $[0, 1]$. We call $\mu_{MC}(S), \mu_{2L}(S)$ the discrete distributions obtained by the Markov chain method and the 2-layer approximation, respectively.

Experimental results: Comparison of distributions

- With $n \leqslant 4$, q is the exact distribution and p is the distribution to be tested. We compute

$$S^4_{KL}(\mu_{MC}) = \sum_{S \in 2^N} \mathbb{D}_{KL}(\mu_{MC}(S)||\boldsymbol{\mu}(S))$$
$$S^4_{KL}(\mu_{2L}) = \sum_{S \in 2^N} \mathbb{D}_{KL}(\mu_{2L}(S)||\boldsymbol{\mu}(S))$$

Experimental results: Comparison of distributions

- With $n \leqslant 4$, q is the exact distribution and p is the distribution to be tested. We compute

$$S^4_{KL}(\mu_{MC}) = \sum_{S \in 2^N} \mathbb{D}_{KL}(\mu_{MC}(S) || \boldsymbol{\mu}(S))$$

$$S^4_{KL}(\mu_{2L}) = \sum_{S \in 2^N} \mathbb{D}_{KL}(\mu_{2L}(S) || \boldsymbol{\mu}(S))$$

- With $n > 4$, we use symmetry properties of the distributions. We compute

$$S^N_{KL}(\mu_{MC}) = \sum_{S,S' \in 2^N \text{ s.t. } |S|=|S'|} \mathbb{D}_{KL}(\mu_{MC}(S) || \mu_{MC}(S'))$$

$$S^N_{KL}(\mu_{2L}) = \sum_{S,S' \in 2^N \text{ s.t. } |S|=|S'|} \mathbb{D}_{KL}(\mu_{2L}(S) || \mu_{2L}(S'))$$

Experimental results: Comparison of distributions

- With $n \leqslant 4$, q is the exact distribution and p is the distribution to be tested. We compute

$$S^4_{KL}(\mu_{MC}) = \sum_{S \in 2^N} \mathbb{D}_{KL}(\mu_{MC}(S)||\boldsymbol{\mu}(S))$$
$$S^4_{KL}(\mu_{2L}) = \sum_{S \in 2^N} \mathbb{D}_{KL}(\mu_{2L}(S)||\boldsymbol{\mu}(S))$$

- With $n > 4$, we use symmetry properties of the distributions. We compute

$$S^N_{KL}(\mu_{MC}) = \sum_{S,S' \in 2^N \text{ s.t. } |S|=|S'|} \mathbb{D}_{KL}(\mu_{MC}(S)||\mu_{MC}(S'))$$
$$S^N_{KL}(\mu_{2L}) = \sum_{S,S' \in 2^N \text{ s.t. } |S|=|S'|} \mathbb{D}_{KL}(\mu_{2L}(S)||\mu_{2L}(S'))$$

- Results

$S^4_{KL}(\mu_{MC})$	$S^4_{KL}(\mu_{2L})$	$S^5_{KL}(\mu_{MC})$	$S^5_{KL}(\mu_{2L})$
0.061	0.059	2.41	2.24

Experimental results: Computation time

Comparison of CPU time (s) for generating 10,000 capacities (3.2 GHz PC with 16 GB of RAM)

Method		$n = 4$	$n = 5$	$n = 6$	$n = 7$
2 layer approximation		2.58	11.51	60.06	330.17
Markov Chain	CPU time	20.46	161.33	$\approx$ 1500	$\approx$ 9000
	T	1170	9000	80,000	500,000

Concluding remarks

- The problem of generating capacities according to a uniform distribution amounts to generate all linear extensions of 2^N

Concluding remarks

- The problem of generating capacities according to a uniform distribution amounts to generate all linear extensions of 2^N
- It is intractable as soon as $n \geqslant 5$

Concluding remarks

- The problem of generating capacities according to a uniform distribution amounts to generate all linear extensions of 2^N
- It is intractable as soon as $n \geqslant 5$
- Naive methods yield poor results

Concluding remarks

- The problem of generating capacities according to a uniform distribution amounts to generate all linear extensions of 2^N
- It is intractable as soon as $n \geqslant 5$
- Naive methods yield poor results
- Good methods try to generate a representative sample of linear extensions: Markov chain method, 2-layer approximation

Concluding remarks

- The problem of generating capacities according to a uniform distribution amounts to generate all linear extensions of 2^N
- It is intractable as soon as $n \geqslant 5$
- Naive methods yield poor results
- Good methods try to generate a representative sample of linear extensions: Markov chain method, 2-layer approximation
- The Markov chain method and the 2-layer approximation method yield similar results, with high accuracy.

Concluding remarks

- The problem of generating capacities according to a uniform distribution amounts to generate all linear extensions of 2^N
- It is intractable as soon as $n \geqslant 5$
- Naive methods yield poor results
- Good methods try to generate a representative sample of linear extensions: Markov chain method, 2-layer approximation
- The Markov chain method and the 2-layer approximation method yield similar results, with high accuracy.
- The 2-layer approximation method is much faster.

Thank you for your attention!