

An Approximation Algorithm for Random Generation of Capacities

Michel Grabisch, Christophe Labreuche, Peiqi Sun

▶ **To cite this version:**

Michel Grabisch, Christophe Labreuche, Peiqi Sun. An Approximation Algorithm for Random Generation of Capacities. Order, 2023, 41 (2), pp.281-306. ⟨10.1007/s11083-023-09630-0⟩. ⟨halshs-04908569⟩

HAL Id: halshs-04908569

https://shs.hal.science/halshs-04908569v1

Submitted on 23 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

An approximation algorithm for random generation of capacities

Michel GRABISCH[1], Christophe LABREUCHE[2] and Peiqi SUN[3*]
[1] Paris School of Economics, Université Paris I - Panthéon-Sorbonne, Paris, France
michel.grabisch@univ-paris1.fr
[2] Thales Research & Technology, Palaiseau, France
christophe.labreuche@thalesgroup.com
[3] Université Paris I - Panthéon-Sorbonne, Paris, France
peiqisun94@gmail.com

February 28, 2023

Abstract

Capacities on a finite set are sets functions vanishing on the empty set and being monotonic w.r.t. inclusion. Since the set of capacities is an order polytope, the problem of randomly generating capacities amounts to generating all linear extensions of the Boolean lattice. This problem is known to be intractable even as soon as $n > 5$, therefore approximate methods have been proposed, most notably one based on Markov chains. Although quite accurate, this method is time consuming. In this paper, we propose the 2-layer approximation method, which generates a subset of linear extensions, eliminating those with very low probability. We show that our method has similar performance compared to the Markov chain but is much less time consuming.

Keywords: capacity, random generation, linear extension, Markov chain

1 Introduction

Capacities, introduced by Choquet [5], are monotone set functions vanishing on the empty set. They are widely used in decision making under risk and uncertainty, multicriteria decision making, and related to many combinatorial problems in Operations Research (see [8] for a detailed account).

In the phase of determining or learning a model based on capacities, it is often important to generate randomly capacities in a uniform way. The problem appears to be surprisingly difficult, due to the monotonicity constraints, and naive methods give poor results, with extremely biased distributions for some coefficients of the capacity. The theoretical exact solution to this problem is however known, since the set of capacities on a finite set N is an order polytope, associated to the Boolean lattice $(2^N, \subseteq)$, and generating uniformly random elements in an order polytope amounts to generating all linear extensions of the underlying poset, that is, in our case, the Boolean lattice. Unfortunately, the number of linear extensions in $(2^N, \subseteq)$ grows tremendously with $|N|$, and is even not known beyond $|N| = 7$. This obliges to resort to approximate methods. The literature has produced many such methods, among which the Markov chain approach seems to give the best results.

The aim of this paper is to propose a new approximate method, counting a limited number of linear extensions, eliminating certain types of linear extensions appearing with very low

*Corresponding author.

probability. The basic idea for this is to concentrate on the two top and two bottom layers on the Boolean lattice, and to eliminate one by one maximal elements in the two top layers, and minimal elements in the two bottom layers. For this reason, the method is called *2-layer approximation method*.

A second contribution of the paper is to provide two ways for measuring the performance of any method generating uniformly distributed random capacities. The first way is based on the probability distribution of the coefficients of the capacity. It is shown that if a capacity μ is uniformly generated, then the distribution of $\mu(S)$ for any $S \subseteq N$ depends only on the cardinality of S, and is symmetric w.r.t. the distribution of $\mu(N \setminus S)$. However, the exact form of the distribution of $\mu(S)$ seems to be very difficult to compute. A second way consists in computing the centroid of the set of generated capacities, and to verify that it is close to the theoretical centroid of the polytope of capacities. Our experiments show that our 2-layer approximation method slightly outperforms the Markov chain method, while taking much less computation time.

The paper is organized as follows. Section 2 settles the problem in its full generality, relating uniform random generation with linear extensions. Section 3 focuses on the generation of capacities and gives the state of the art. The 2-layer approximation method is developed in Section 4 in full detail. Section 5 is devoted to the measure of performance of any method of random generation of capacities.

2 Preliminaries

Let P be a finite set, endowed with a partial order $\preccurlyeq$, that is, a reflexive, antisymmetric and transitive binary relation. We say that $(P, \preccurlyeq)$ is a *(finite) poset*. Recall that $x \in P$ is *maximal* if $x \preccurlyeq y$ with $y \in P$ implies $x = y$. Similarly, $x \in P$ is *minimal* if $y \preccurlyeq x$ with $y \in P$ implies $x = y$. We denote by $\mathsf{Max}(P, \preccurlyeq)$ and $\mathsf{Min}(P, \preccurlyeq)$ (simply $\mathsf{Max}(P), \mathsf{Min}(P)$ if there is no ambiguity) the set of maximal and minimal elements of P, respectively.

The *order polytope* [15] associated to $(P, \preccurlyeq)$, denoted by $\mathcal{O}(P)$, is the set

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

It is a polytope of dimension $p := |P|$, whose volume $V(\mathcal{O}(P))$ can be computed by the following formula [15]:

$$V(\mathcal{O}(P)) = \frac{e(P)}{p!}, \tag{1}$$

where $e(P)$ is the number of linear extensions of $(P, \preccurlyeq)$. A *linear extension* of $(P, \preccurlyeq)$ is a total order $\leqslant$ on P which is compatible with the partial order $\preccurlyeq$ in the following sense: $x \preccurlyeq y$ implies $x \leqslant y$. It is convenient to denote a linear extension by the sequence of the elements of P arranged in increasing order according to $\leqslant$: assuming $P = \{x_1, \ldots, x_p\}$, the linear extension $\leqslant$ is denoted by $x_{\sigma(1)}, \ldots, x_{\sigma(p)}$, with σ a permutation on $\{1, \ldots, p\}$ and $x_{\sigma(1)} < \cdots < x_{\sigma(p)}$.

Let $N := \{1, 2, \ldots, n\}$ be a finite set of n elements. A *(normalized) capacity* [5, 16, 8] on N is a set function $\mu : 2^N \longrightarrow [0,1]$ satisfying $\mu(\varnothing) = 0$, $\mu(N) = 1$ (normalization), and the property $S \subseteq T \Rightarrow \mu(S) \leqslant \mu(T)$ (monotonicity).

We denote the set of capacities on N by $\mathcal{C}(N)$. From its definition, one can see that $\mathcal{C}(N)$ is an order polytope, whose underlying poset is $(2^N \setminus \{\varnothing, N\}, \subseteq)$.

3 Random generation of capacities

3.1 Random generation and linear extensions

The problem of randomly generating capacities according to a uniform distribution amounts to picking a point in the polytope $\mathcal{C}(N)$ in a uniform way. This is made simple because $\mathcal{C}(N)$ is an order polytope. According to Stanley [15], given a poset $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$ and its associated order polytope $\mathcal{O}(P)$, each linear extension of $(P, \preccurlyeq)$ defines a region in $\mathcal{O}(P)$:

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

where σ is the permutation associated to the linear extension, i.e., $x_{\sigma(1)} < \cdots < x_{\sigma(p)}$. All regions R_σ are identical (up to a change of coordinates), are p-dimensional simplices with volume $\frac{1}{p!}$, they cover the polytope and have disjoint interiors, and this leads to (1). Vertices of this simplex are the $p+1$ functions given by

$$0 = f(x_{\sigma(1)}) = \cdots = f(x_{\sigma(k)}),\ f(x_{\sigma(k+1)}) = \cdots = f(x_{\sigma(p)}) = 1, \quad (k = 1, \ldots, p-1) \qquad (2)$$

and the two constant functions 0 and 1.

As a consequence, the random generation of a point in $\mathcal{O}(P)$ w.r.t. a uniform distribution amounts to uniformly selecting a linear extension, and then to uniformly selecting a point in the associated region R_σ. The latter step is done as follows: generate p independent numbers in $[0, 1]$ according to the uniform distribution, then order them in increasing order, say, $z_1 \leqslant \cdots \leqslant z_p$, and put $f(x_{\sigma(1)}) = z_1, \ldots, f(x_{\sigma(p)}) = z_p$. This define $f \in \mathcal{O}(P)$.

The above technique can be applied to the uniform generation of capacities. As explained above, $(2^N \setminus \{\varnothing, N\}, \subseteq)$ is the underlying poset. It suffices to select randomly a linear extension $<$ of $(2^N \setminus \{\varnothing, N\}, \subseteq)$, say

$$S_{\sigma(1)} < \cdots < S_{\sigma(2^n-2)},$$

and then generate independently and uniformly $2^n - 2$ numbers in $[0, 1]$, letting $\mu(S_{\sigma(i)}) = z_i$, where z_i is the ith smallest generated number. The problem is then theoretically solved but the above method reveals to be intractable as soon as $n \geqslant 5$. Indeed, the number of linear extensions on the Boolean lattice $(2^N, \subseteq)$ (note that it is equal to the number of linear extensions on $(2^N \setminus \{\varnothing, N\}, \subseteq)$) increases extremely fast, as indicated in Table 1 below. There is no known

n	$e(2^N)$
1	1
2	2
3	48
4	1680384
5	14807804035657359360
6	1413779116972278871171959703162007956302054769577164 80

Table 1: Number of linear extensions of 2^N

closed-form formula for computing the number of linear extensions on 2^N, nor is it known beyond $n = 7$ (it is sequence A046873 in the *Online Encyclopedia of Integer Sequences*), however bounds are known (see, e.g., [2]). More generally, the problem of counting the number of linear extensions on a poset is a $\sharp$-P complete problem, as shown in [3].

As a consequence, only approximate methods can be used to generate capacities in a uniform way.

3.2 Related literature

We give here a brief account of the literature on approximate methods for generating capacities. One of the simplest method (shown in [9]) is to select randomly in each step one element of a capacity and then draw uniformly a number within an interval imposed by the monotonicity constraints of the already drawn numbers. This method, called *random node generator*, although very simple to implement, yields fairly biased results.

Several methods try to generate linear extensions of the considered poset, from which it is immediate to generate a capacity as we explained above. Karzanov and Khachiyan [10] propose to use the Markov Chain technique to generate random linear extensions of a given poset P. Consider each linear extension as a state of a Markov Chain, and denote by E the set of all linear extensions of P. The Markov chain with its transition probability matrix describes a random walk through the simplices R_σ of the order polytope $\mathcal{O}(P)$, and Karzanov and Khachiyan prove that its transition probability is an ergodic time-reversible Markov chain with a uniform stationary distribution, which means that for an arbitrary initial probability distribution on E, after T steps (T large enough), the distribution converges to the uniform distribution on E. Thus, for any given poset P and sufficiently large T, the Markov chain method gives a nearly uniform generator of linear extensions of the poset. However, when the cardinality of P is large, it is costly in computation time. In addition to the original classical Karzanov-Khachiyan chain method, there exist some more efficient Markov Chain algorithms, for example, Bubley and Dyer introduced the Insertion Chain algorithm in [4], while [17] mentions some other Markov Chain algorithms.

Combarro et al. [6] gives a good survey of methods of generating capacities, and propose a method called Minimals to generate linear extensions of P, which consists in selecting a minimal element of P with a certain probability, then delete it and repeat this procedure until all elements have been selected. If we could determine this probability exactly, we would obtain an exact method. For example, for generating 2-symmetric capacities, Miranda and Garcia Segador [13] transform the problem into generating a linear extension of the Young rectangular diagram. Since there exists a closed-form formula for the number of linear extensions of the Young diagram, according to the general probability formula of an element x being selected $\mathbb{P}(x) = \frac{e(P\backslash x)}{e(P)}$ (see Formula (3) below), we get an exact method to generate 2-symmetric capacities. Besides, Beliakov introduces an approximation method in [1] to generate supermodular capacities, and Miranda and Garcia Segador introduce an exact method in [14] to generate 2-additive capacities.

For a general capacity, the idea of generating a linear extension by the method introduced in [6] could be used, at the condition to find a way to determine the probability of selecting an element. Miranda and Garcia Segador have proposed some approximation methods to obtain this probability, namely the Bottom-up method [12], while Combarro et al. have proposed the Minimals Plus method [7]. However, the Bottom-up method can be applied only to a restricted family of posets. For example, it can be applied to the Boolean lattice 2^N till $|N| = 3$, but beyond this, the probability of selecting certain elements could be 0, making the method inapplicable. Minimals Plus combines the idea of the Minimals method and the Markov Chain algorithm, and it also uses a new algorithm to calculate the probability of selecting each element. It could be also seen as an improvement of the Bottom-Up method. When Bottom-Up is applicable for a poset, Minimals Plus is an exact method with a lower computational cost. For any other poset, Minimals Plus needs to be combined with a Markov Chain algorithm, which considerably increases the computational cost.

In this article, we present another approximation method to determine the probability of selecting a minimal (or maximal) element, which is presented in the next section.

4 The 2-layer approximation algorithm

4.1 Basic idea

Let us consider an arbitrary poset $(P, \preccurlyeq)$ with $P = \{x_1, \ldots, x_p\}$. Clearly, any linear extension $x_{\sigma(1)}, \ldots, x_{\sigma(p)}$ starts with a minimal element and terminates with a maximal element, i.e., $x_{\sigma(1)} \in \mathsf{Min}(P)$ and $x_{\sigma(p)} \in \mathsf{Max}(P)$. Observe that $x_{\sigma(2)}$ must be a minimal element of $P \backslash \{x_{\sigma(1)}\}$, and $x_{\sigma(p-1)}$ is a maximal element on $P \setminus \{x_{\sigma(p)}\}$, and so on. This gives a recursive structure to linear extensions.

Based on this observation, the probability that a linear extension of P starts with $m \in \mathsf{Min}(P)$ is

$$\begin{aligned} \Pr(m \mid P) &= \frac{e(P \setminus \{m\})}{e(P)} \\ &= 1 - \frac{\sum_{m' \in \mathsf{Min}(P), m' \neq m} e(P \setminus \{m'\})}{e(P)}, \end{aligned} \tag{3}$$

and similarly the probability that a linear extension terminates with $M \in \mathsf{Max}(P)$ is

$$\begin{aligned} \Pr(M \mid P) &= \frac{e(P \setminus \{M\})}{e(P)} \\ &= 1 - \frac{\sum_{M' \in \mathsf{Max}(P), M' \neq M} e(P \setminus \{M'\})}{e(P)}. \end{aligned} \tag{4}$$

Then, generating a linear extension amounts to choosing, according to the correct probability given by (3) and (4), a minimal or a maximal element of a poset which is diminished by one element at each step. As the computation of the probability directly depends on the number of linear extensions, the computation can be exact only when P becomes small enough. Otherwise, some approximation must be done. The idea we propose is to 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. These two subposets are also used to make the computation of the respective probabilities by (3) and (4). The next section gives the details of our approximation method, where P is now a subset of the Boolean lattice $(2^N, \subseteq)$.[1]

4.2 Approximation method

From now on, we deal with posets which are particular subsets of $(2^N \setminus \{\varnothing, N\}, \subseteq)$. As explained in Section 4.1, we proceed by deleting step by step minimal or maximal elements of the Boolean lattice $(2^N \setminus \{\varnothing, N\}, \subseteq)$, which yields at each step of the process a poset $(H, \subseteq)$ which is a subset of $(2^N \setminus \{\varnothing, N\}, \subseteq)$. For convenience, and referring to its Hasse diagram, we call *nodes* the subsets of N in H, denoting them most often by $x, y, \ldots$ (unless the usual set notation $S, T, \ldots$ is more adequate), and we define a *layer* of H as the set of all nodes (subsets) of same cardinality.

The principle of the method is to limit the search of maximal elements (resp., minimal elements) of H as well as the computation of their probability of occurrence to the two top layers (resp., to the two bottom layers). We denote by T_H the poset formed by the two top layers of H, and for ease of readability often write $T_H[h, k, |I|]$ to specify its *basic parameters* $h, k, |I|$

[1] As any poset P can be seen as a subposet of a Boolean lattice, our method can then be applied to any poset P (and hence not limited to the generation of capacities), up to some rewriting.

explained below. The upper layer contains h nodes of cardinality say ℓ, with $2 \leqslant \ell \leqslant n-1$, while the lower layer contains k nodes of cardinality $\ell - 1$, among which some of them are isolated, i.e., with no predecessor, and we denote by I the set of isolated nodes and by $|I|$ its number. Note that during the deletion process, it may happen that isolated nodes appear in the upper layer, but this is of no incidence on the process. Therefore their number is not a basic parameter, and referring in the sequel to isolated nodes implicitely means those of the lower layer. A node y of the lower layer has *predecessors* in the upper layer, i.e., sets x satisfying $x \supset y$. We denote by $\mathsf{prec}(y)$ the set of predecessors of y. Similarly, for a node x of the upper layer, we denote by $\mathsf{succ}(x)$ the set of its *successors*, i.e., subsets of x in the lower layer. In a dual way, we introduce B_H (denoted also $B_H[h', k', |I'|]$), the poset of the two bottom layers of H, with h' nodes on the upper level, k' nodes on the lower level, and I' is the set of isolated nodes in the upper (!) level. Fig. 1 illustrates the definitions.

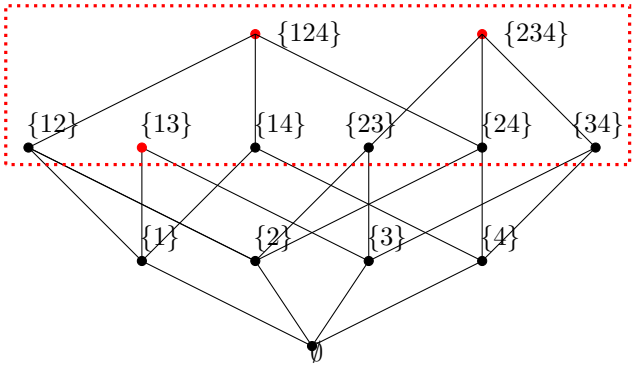


Figure 1: Example of poset H with the poset $T_H(2, 6, 1)$ formed by the two top layers framed in red. Braces and commas are omitted for denoting sets. Maximal elements of H are in red. 13 is the only isolated node.

Consider $H \subseteq 2^N \setminus \{\varnothing, N\}$ and take a maximal element M of H belonging to T_H (i.e., a maximal element of T_H), and a minimal element m of H belonging to B_H (note that it is equivalent to write $T_H \setminus \{M\}$ or $T_{H\setminus\{M\}}$). The probabilities that M and m are terminal and initial elements of a linear extension of H are approximated by:

$$\Pr(M \mid H) \approx \frac{e(T_H \setminus \{M\})}{e(T_H)} = \Pr(M \mid T_H) \tag{5}$$

$$\Pr(m \mid H) \approx \frac{e(B_H \setminus \{m\})}{e(B_H)} = \Pr(m \mid B_H). \tag{6}$$

This approximation is based on the following two observations:

(i) The probability that a node M is maximal in H when M does not belong to T_H is approximately zero. This is because the number of nodes which are above a given node in layer k increases very fast with k. Indeed, supposing that a node has in average ℓ predecessors, each predecessor has in turn ℓ predecessors as well, so that the number of nodes above a given one in the two layers above it is of the order ℓ^2, and so on. Consequently, a node x in the third layer (or below) has very little probability to become a maximal element and hence to be selected, since all nodes above it must be eliminated first, *without eliminating all nodes of the 1st layer*, because otherwise x becomes a node of the 2nd layer.

If a general expression for this probability seems to be out of reach, we can have an exact value for $n = 4$. Observe that to pick $\{1\}$ of the 3d layer as a maximal element obliges,

after having selected $\{1,2,3\}$, $\{1,2,4\}$, $\{1,3,4\}$ and their subsets on the 2nd layer, to leave $\{2,3,4\}$ unselected (otherwise the 1st layer becomes empty and $\{1\}$ becomes a node of the 2nd layer). In other words, the only linear extensions corresponding to this situation are those where (listed in increasing order) $\{1\}$ is ranked after $\{2,3,4\}$. Similarly, the same arises with $\{2\}$ ranked after $\{1,3,4\}$, $\{3\}$ ranked after $\{1,2,4\}$, and $\{4\}$ ranked after $\{1,2,3\}$. We have denumbered 9216 such linear extensions, on a total of 1680384 linear extensions, yielding a probability equal to 0.00548, which is very small.

(ii) Our method selects and removes nodes only in the two top and two bottom layers, leaving the other layers untouched. Thus, the number of linear extensions in T_H finishing by $M \in T_H$ should differ from the number of linear extensions in H finishing by M only by a multiplicative constant, independent from M.

The algorithm for the generation of linear extensions with uniform probability is described below.

generate-linext(P, l)
Input: a poset P subset of $2^N \setminus \{\varnothing, N\}$
Output: a linear extension l of P generated with an approximately 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 $\Pr(M \mid T_H[h,k,|I|])$
        Add M at the beginning of $lmax$
        $H \leftarrow H \setminus \{M\}$
    otherwise
        Select $m \in B_H[h',k',|I'|]$ with probability $\Pr(m \mid B_H[h',k',|I'|])$
        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

Observe the particularity when the height of H is equal to 2. Indeed, the two top layers coincide with the two bottom layers, so that one may either select maximal or minimal elements. In order to minimize the number of steps, H is considered as B_H if there are more nodes in the upper layer, and as T_H otherwise.

We now aim at getting an explicit expression of $\Pr(M \mid T_H[h,k,|I|])$ (the same development can be done in a dual way for $\Pr(\mid B_H[h',k',|I'|])$).

We start with a simple property and observation.

Lemma 1. For any poset $H \subseteq 2^N \setminus \{\varnothing, N\}$, the number of linear extensions in the poset $T_H[h,k,|I|]$ is

$$e(T_H(h,k,|I|)) = e(T_{H\setminus I}[h,k-|I|,0]) \times \prod_{i=1}^{|I|}(k-|I|+h+i).$$

Proof. Take $H \subseteq 2^N \setminus \{\varnothing, N\}$ and consider the two top layers $T_H[h,k,|I|]$. The isolated nodes being incomparable with the remaining elements, they can appear at any position in a linear extension of $T_H[h,k,|I|]$. Hence, there are $k+h-|I|+1$ possibilities to insert one isolated node into a linear extension of $T_{H\setminus I}[h,k-|I|,0]$, then $k+h-|I|+2$ possibilities to insert a second one independently of the first choice, etc. This yields the desired formula. □

Deriving other properties of $e(T_H)$ requires some assumption on the structure of T_H. The following definition is central in our development.

Definition 1. Let x be a node of the upper layer of T_H.

(i) The function f_x assigns to every node y of the lower layer an integer as follows:

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

(ii) 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.

Definitions for B_H are dual: x is in the lower layer and for every y in the upper layer $f_x(y)$ is the number of successors of y, etc.

Definition 2. We say that T_H (respectively, B_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 (respectively, in the lower layer).

Example 1. Consider the poset given in Figure 2. Let us compute the function f_x for this poset. We find:

	$y=\{1,2\}$	$y=\{1,4\}$	$y=\{2,3\}$	$y=\{2,4\}$	$y=\{3,4\}$
$f_{\{1,2,4\}}(y)$	1	1	0	2	0
$f_{\{2,3,4\}}(y)$	0	0	1	2	1

From this, we compute the function n_x. We find

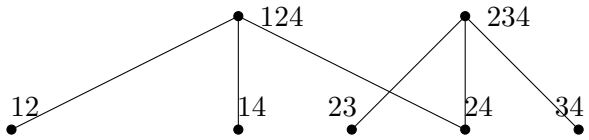


Figure 2: Example of a regular poset T_H

	$r=0$	$r=1$	$r=2$
$n_{\{1,2,4\}}(r)$	2	2	1
$n_{\{2,3,4\}}(r)$	2	2	1

We see that $n_{\{1,2,4\}}(r) = n_{\{2,3,4\}}(r)$ for every r, therefore this poset is regular.

As regularity will be the key property in the following, we elaborate on it and relate it to more intuitive properties. The first one is based on the following observation.

Observation 1. For any two nodes x, x' in the upper level of poset T_H, we always have $|\mathsf{succ}(x) \cap \mathsf{succ}(x')| \leqslant 1$.

Proof. Let x be in the upper layer of T_H and suppose that x has at least two successors. As successors of x have one less element than x, two successors of x have the form $x \setminus \{i\}$ and $x \setminus \{j\}$, for distinct $i, j \in x$. If they are contained in another subset y, this implies $x \setminus \{i\} \subset y$, $x \setminus \{j\} \subset y$, which in turn implies $x \subset y$, a contradiction with $|x| = |y|$. □

Definition 3. (i) We say that T_H is *closed under intersection* if for every two nodes x, x' of the upper layer, $|\mathsf{succ}(x) \cap \mathsf{succ}(x')| = 1$. i.e., $x \cap x'$ belongs to the lower layer. Similarly for the two bottom layers, B_H is *closed under union* if for every node x, x' of the lower layer, $x \cup x'$ belongs to the upper layer.[2]

(ii) We say that T_H is *balanced* if $|\mathsf{succ}(x)| = |\mathsf{succ}(x')|$ for every x, x' in the upper layer. Similarly, B_H is balanced if $|\mathsf{prec}(x)| = |\mathsf{prec}(x')|$ for every x, x' in the lower layer.

Example 1 continued. The poset in Fig.2 is closed under intersection since $\mathsf{succ}(\{1,2,4\}) \cap \mathsf{succ}(\{2,3,4\}) = \{2,4\}$, and also balanced since $|\mathsf{succ}(\{1,2,4\})| = |\mathsf{succ}(\{2,3,4\})|$.

The next proposition relates all these properties.

Proposition 1. Consider a poset H.

(i) If T_H is regular, then it is balanced.

(ii) If T_H is closed under intersection and balanced, then it is regular.

(iii) Regularity of T_H does not imply that T_H is closed under intersection.

(iv) T_H is regular if $H = 2^N \setminus \{N, \varnothing\}$.

(v) Suppose $n \geqslant 3$ and for every node x in the upper layer, $|x| = |\mathsf{succ}(x)| = n - 1$. Then T_H is closed under intersection and regular.

The same assertions hold for B_H.

[2]See the Appendix for a characterization of posets closed under intersection.

(see proof in the Appendix)

Thus, in the algorithm **generate-linext**, the alternance of selecting maximal elements and minimal elements permits to keep as long as possible the assumption of regularity.

We are now in position to show the next important result.

Proposition 2. Consider a poset H with its two top layers $T_H[h,k,0]$ without isolated nodes (in the lower layer), and suppose that it is regular. Then, for every x in the upper layer,

$$e(T_H[h,k,0]) = h \times e(T_{H\setminus\{x\}}[h-1,k,|I|]), \tag{7}$$

where I is the set of isolated nodes in the (lower layer of the) two top layers of $H \setminus \{x\}$.

Proof. Observe that $H \setminus \{x\}$ has a number $|I|$ of isolated nodes which does not depend on x because $n_x(1) = n_{x'}(1)$ for every x, x' in the upper layer. Moreover, as n_x is invariant w.r.t x, every node x has the same structure of successors so that $e(T_{H\setminus\{x\}}[h-1,k,|I|])$ does not depend on x. As any node of the upper layer is a maximal element, i.e., can be the last element in a linear extension, this proves (7). □

The next step is to establish the probability to select a maximal element from T_H. A maximal element can be either a node from the upper layer, or an isolated node of the lower layer. It is easy to see that in the latter case, the probability does not depend on the isolated element. This is because an isolated node can appear in any position of the linear extension, without constraint. Interestingly, one can prove that under regularity, the same property holds for nodes of the upper layer.

Proposition 3. Consider a poset H with its two top layers T_H, and suppose that it is regular. Then for every x, x' in the upper layer, $\Pr(x \mid T_H) = \Pr(x' \mid T_H)$, where $\Pr(x \mid T_H)$ is the probability that a linear extension in T_H terminates by x.

Proof. As observed in the proof of Proposition 2, for every x, x' in the upper layer, $T_H \setminus \{x\}$ and $T_H \setminus \{x'\}$ are isomorphic by regularity. Hence the result holds. □

Remark that the same property when removing a node from the bottom layer holds for the two bottom layers, with symmetric conditions.

As a consequence of the above proposition and remark, we can get an explicit expression of these probabilities.

Proposition 4. 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 an isolated node of the lower layer terminates a linear extension are given by

$$\mathbb{P}_u(T_H[h,k,|I|]) = \frac{1}{h}\frac{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i)}{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i) + |I| \times \prod_{i=1}^{|I''|}(h-1+k-|I'|+i)\prod_{i=1}^{|I|-1}(h+k-|I|+i)} \tag{8}$$

$$\mathbb{P}_l(T_H[h,k,|I|]) = \frac{\prod_{i=1}^{|I''|}(h-1+k-|I'|+i)\prod_{i=1}^{|I|-1}(h+k-|I|+i)}{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i) + |I| \times \prod_{i=1}^{|I''|}(h-1+k-|I'|+i)\prod_{i=1}^{|I|-1}(h+k-|I|+i)}, \tag{9}$$

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

Proof. Removing a node x of the upper layer of $T_H[h,k,|I|]$ yields the poset $T_{H\setminus\{x\}}[h-1,k,|I'|]$ with $I' \supseteq I$. Then by Lemma 1

$$e(T_{H\setminus\{x\}}[h-1,k,|I'|]) = e(T_{H\setminus(\{x\}\cup I')}[h-1,k-|I'|,0]) \times \prod_{i=1}^{|I'|}(h-1+k-|I'|+i). \quad (10)$$

Removing an isolated node y of the lower layer yields the poset $T_{H\setminus\{y\}}[h,k-1,|I|-1]$. Removing all isolated nodes and one node x of the upper layer yields the poset $T_{H\setminus(I\cup\{x\})}[h-1,k-|I|,|I''|]$, with the disjoint union $I \cup I'' = I'$, i.e., $|I| + |I''| = |I'|$. Applying again Lemma 1 we obtain

$$\begin{aligned}
& e(T_{H\setminus\{y\}}[h,k-1,|I|-1] \\
&= e(T_{H\setminus I}[h,k-|I|,0] \prod_{i=1}^{|I|-1}(h+k-|I|+i) \\
&= h \times e(T_{H\setminus(I\cup\{x\})}[h-1,k-|I|,|I''|] \prod_{i=1}^{|I|-1}(h+k-|I|+i) \\
&= h \times e(T_{H\setminus(I'\cup\{x\})}[h-1,k-|I'|,0] \prod_{i=1}^{|I''|}(h-1+k-|I'|+i) \prod_{i=1}^{|I|-1}(h+k-|I|+i) \quad (11)
\end{aligned}$$

where in the second equality we have used the fact that $T_{H\setminus I}[h,k-|I|,0]$ is regular, as well as Proposition 2.

By (4), we have for any node x of the upper level and any isolated node y:

$$\begin{aligned}
& \mathbb{P}_u(T_H[h,k,|I|]) \\
&= \frac{e(T_{H\setminus\{x\}}[h-1,k,|I'|])}{e(T_H[h,k,|I|])} \\
&= \frac{e(T_{H\setminus\{x\}}[h-1,k,|I'|])}{h \times e(T_{H\setminus\{x\}}[h-1,k,|I'|]) + |I| \times e(T_{H\setminus\{y\}}[h,k-1,|I|-1])} \\
&= \frac{1}{h} \frac{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i)}{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i) + |I| \times \prod_{i=1}^{|I''|}(h-1+k-|I'|+i) \prod_{i=1}^{|I|-1}(h+k-|I|+i)}
\end{aligned}$$

where in the second equality we have used Proposition 3, and in the third equality Eqs. (10) and (11). Similarly, we get

$$\begin{aligned}
& \mathbb{P}_l(T_H[h,k,|I|]) \\
&= \frac{e(T_{H\setminus\{y\}}[h,k-1,|I|-1])}{h \times e(T_{H\setminus\{x\}}[h-1,k,|I'|]) + |I| \times e(T_{H\setminus\{y\}}[h,k-1,|I|-1])} \\
&= \frac{\prod_{i=1}^{|I''|}(h-1+k-|I'|+i) \prod_{i=1}^{|I|-1}(h+k-|I|+i)}{\prod_{i=1}^{|I'|}(h-1+k-|I'|+i) + |I| \times \prod_{i=1}^{|I''|}(h-1+k-|I'|+i) \prod_{i=1}^{|I|-1}(h+k-|I|+i)}.
\end{aligned}$$

□

As a conclusion,

$$\Pr(M \mid T_H[h,k,|I|]) = \begin{cases} \mathbb{P}_u(T_H[h,k,|I|]), & \text{if } M \text{ is a node of the upper layer} \\ \mathbb{P}_l(T_H[h,k,|I|]), & \text{if } M \text{ is an isolated node of the lower layer.} \end{cases} \quad (12)$$

Similar expressions can be obtained for $\Pr(m \mid B_H(h',k',|I'|))$. Both can be used in Algorithm **generate-linext**. Note that these expressions of probabilities are approximative, as the assumption that both $T_H(h,k,|I|)$ and $B_H(h',k',|I'|)$ are regular are valid in the first iteration, but not necessarily in the subsequent iterations.

4.3 Complexity analysis

The algorithm **generate-linext**(P, l) iterates over the subsets in $2^N \setminus \{\varnothing, N\}$ until all coalitions are picked up, and thus is composed of $2^n - 2$ iterations. Each iteration considers T_H or B_H. As these two situations yield the same complexity, let us consider T_H. The cardinality of the upper layer is denoted by ℓ. We first compute the probability that a non-dominated node is selected. To this end, we only need to compute the probability $\mathbb{P}_u(T_H[h, k, |I|])$ of the upper nodes and the probability $\mathbb{P}_l(T_H[h, k, |I|])$ of the lower nodes. By (8) and (9), the computation of these two probabilities requires $|I''| \times (|I| - 1) + |I'|$. As $|I|, |I'|, |I''| \leq \left(\begin{array}{c} n \\ l-1 \end{array} \right) \leq 2^n$, these computations take less than $O(4^n)$ operations. Selecting the non-dominated elements requires to sweep at most $h + k \leq \left(\begin{array}{c} n \\ l-1 \end{array} \right) + \left(\begin{array}{c} n \\ l \end{array} \right) \leq 2^n$ elements. Hence, selecting a non-dominated element of T_H has complexity at most $O(4^n)$. As a consequence, the complexity of **generate-linext**(P, l) is in total at most $O(8^n)$.

The complexity of the Markov chain algorithm is for generating linear order of size r is $O(r^5 \log\, r)$ [10]. In our case ($r = 2^n$), this yields a complexity $O(n\, 32^n)$ to be compared to $O(8^n)$.

4.4 Example

We illustrate the 2-layer approximation algorithm by applying it for generating a linear extension with $n = 4$ (see Figure 3 (a), (b), ... (i)). For simplicity, sets are denoted by 123 instead of $\{1, 2, 3\}$, etc.

- (a): the two top layers are considered. Here, since all maximal nodes are on the first layer, by Proposition 3, all maximal nodes have the same probability of 0.25. Suppose 234 is selected, which yields Figure (b).
- (b): now the two bottom layers are considered. Similarly, all minimal nodes are equiprobable. Suppose node 1 is selected, this yields (c).
- (c): As for (a), we consider again the two top layers. All maximal nodes are on the first layer, hence equiprobable. Supposing 134 is selected, (d) is obtained.
- (d): As for (b), we consider the two bottom layers, whose minimal nodes are on the bottom layer, hence equiprobable. Supposing node 4 is selected, we obtain (e).
- (e): In the two upper layers, one maximal node is on the second layer. Then Proposition 4 has to be used to compute the probabilities for the first and second layer, which are respectively 0.437 and 0.125. Supposing 124 is selected, (f) is obtained.
- (f): In the two bottom layers, one minimal element is 14 on the second layer, hence again Proposition 4 must be used. One finds 0.437 for nodes 2 and 3, and 0.125 for 14. Supposing 14 is selected, we obtain (g).
- (g): Now the probability of selecting 123 is 0.67, and is 0.167 for selecting 24 and 34. Supposing 123 is selected, we obtain (h).
- (h): The two top and bottom layers coincide. We consider them as the two bottom layers, since there are less minimal elements than maximal elements, and select a minimal element. They are equiprobable. Supposing node 2 is selected, (i) is obtained.

- (i): Again, we select a minimal node. Applying Proposition 4, the probabilities are 0.67 for selecting node 3, and 0.167 for selecting 12 and 24. Supposing node 3 is selected, we obtain (j).
- (j): there is only one layer, i.e., all nodes are both maximal and minimal and are equiprobable. We may suppose that they are selected in this order: 12, 34, 24, 23, 13.

Finally, the obtained linear extension is: 1, 4, 14, 2, 3, 12, 34, 24, 23, 13, 123, 124, 134, 234. Observe that from (a) to (j), the regularity assumption is satisfied. Indeed, at each step, nodes of the upper layer of T_H (resp., lower layer of B_H) have 3 successors (resp., 3 predecessors), so that by Proposition 2 (v), T_H and B_H are regular.

5 Measures of performance

An important point is to be able to assess the performance of a given algorithm generating capacities in a uniform way, i.e., how uniform is the distribution obtained? Given the high dimensionality of the polyhedron of capacities, no graphical representation is possible, and the task reveals to be a challenging one. In what follows, we propose two ways of measuring the performance, based on the distribution of $\mu(S)$ for a given subset S, and on the centroid of $\mathcal{C}(N)$.

5.1 Distribution of $\mu(S)$

Uniform distribution of the capacity does not mean that the distribution of $\mu(S)$ for any $S \subseteq N$, $S \neq \varnothing, N$ is uniform. An explicit and convenient expression for the distribution of $\mu(S)$ seems however very difficult to obtain. The following can be said, however.

We denote by $\boldsymbol{\mu}$ the corresponding random variable. Referring to the notation of Section 3.1, consider a linear extension associated to the permutation σ on $2^N \setminus \{\varnothing, N\}$, and R_σ the corresponding simplex in $\mathcal{C}(N)$. Supposing $\boldsymbol{\mu} \in R_\sigma$, we have 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:

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

with $\mathrm{Beta}(r,s) = \frac{1}{B(r,s)}u^{r-1}(1-u)^{s-1}$, and $B(r,s)$ is the Beta function

$$B(r,s) = \int_0^1 t^{r-1}(1-t)^{s-1}\mathrm{d}t = \frac{\Gamma(s)\Gamma(r)}{\Gamma(r+s)}$$

with $\Gamma(r) = \int_0^1 t^{r-1}e^{-t}\mathrm{d}t$ the Gamma function. 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

$$\begin{aligned} F_{\boldsymbol{\mu}(S)}(\alpha) = \Pr(\boldsymbol{\mu}(S) \leqslant \alpha) &= \sum_{\sigma \in E(2^N \setminus \{\varnothing, N\})} \Pr(\boldsymbol{\mu}(S) \leqslant \alpha \mid \boldsymbol{\mu} \in R_\sigma)\Pr(\boldsymbol{\mu} \in R_\sigma) \\ &= \frac{1}{e(2^N)} \sum_{\sigma \in E(2^N \setminus \{\varnothing, N\})} \Pr(\boldsymbol{\mu}(S) \leqslant \alpha \mid \boldsymbol{\mu} \in R_\sigma) \\ &= \frac{1}{e(2^N)} \sum_{\sigma \in E(2^N \setminus \{\varnothing, N\})} \mathrm{OS}_{k(S,\sigma)}(\alpha), \end{aligned} \tag{13}$$

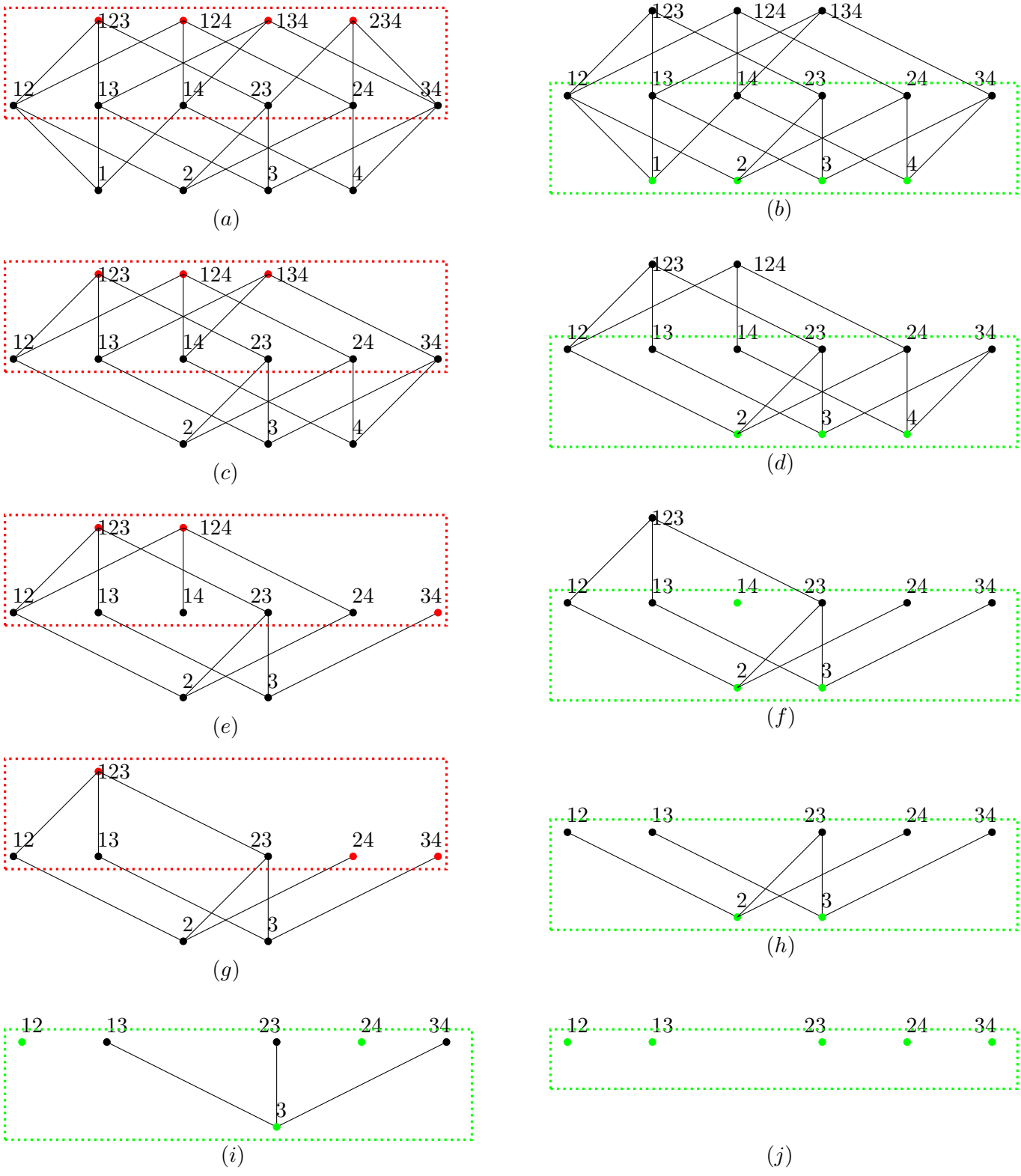


Figure 3: Process of node selection according to Algorithm **generate-linext**. The dotted red and green boxes indicate the two top and bottom layers under consideration, respectively. Red nodes are maximal elements in the two top layers, while green nodes are minimal elements in the two bottom layers.

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

Based on this formula, the following can be shown.

Lemma 2. Assume $\boldsymbol{\mu}$ is uniformly distributed and take $\varnothing \neq S, S' \subset N$. Then

(i) $\boldsymbol{\mu}(S)$ and $\boldsymbol{\mu}(S')$ for $|S| = |S'|$ are identically distributed.

(ii) $\boldsymbol{\mu}(S)$ and $1 - \boldsymbol{\mu}(N \setminus S)$ are identically distributed.

Proof. (i) Consider distinct subsets $S, S' \in 2^N \setminus \{\varnothing, N\}$ such that $|S| = |S'|$, and fix a linear extension σ. Then there exists a permutation π on N such that $\pi(S) = S'$, and $k(S, \sigma) = k(\pi(S), \pi(\sigma))$, where $\pi(S) = \{\pi(i), i \in S\}$ and $\pi(\sigma)$ is the linear extension $\pi(S_{\sigma(1)}), \ldots, \pi(S_{\sigma(2^n-2)})$. Applying (13) we find:

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

the last equation following from the fact that π is a bijection on $E(2^N \setminus \{\varnothing, N\})$.

(ii) Consider the conjugate capacity $\overline{\boldsymbol{\mu}}$ defined by $\boldsymbol{\mu}(S) = 1 - \boldsymbol{\mu}(N \setminus S)$. Taking a linear extension $S_\sigma(1) < S_{\sigma(2)} < \cdots < S_{\sigma(2^n-2)}$, it is easy to see that the sequence

$$N \setminus S_{\sigma(2^n-2)}, \ldots, N \setminus S_{\sigma(2)}, N \setminus S_{\sigma(1)}$$

is also a linear extension $\overline{\sigma}$ with the property $k(S, \sigma) = 2^n - 2 - k(N \setminus S, \overline{\sigma})$. Using the fact that OS_k is distributed as $1 - \mathrm{OS}_{2^n-2-k}$, we obtain

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

the last equation following from the fact that $\sigma \mapsto \overline{\sigma}$ is a bijection on $E(2^N \setminus \{\varnothing, N\})$.

□

5.2 The centroid

In order to check if the set of generated capacities is well balanced in the whole polyhedron $\mathcal{C}(N)$, we propose to compute its average capacity and to check if it coincides with the centroid of $\mathcal{C}(N)$. We recall that for a convex polyhedron P, the centroid can be obtained from a triangulation of the polyhedron into simplices, provided they have the same volume. The centroid of a simplex is the barycenter of its vertices, and the centroid of P is the average of the centroids of all simplices. Note that it differs from the barycenter of the vertices of P in general.

From Section 3.1, we know that $\mathcal{C}(N)$ can be triangulated by the regions R_σ, each region corresponding to a linear extension, and its vertices are given by Eq. (2).

We illustrate the method of computation with $n = 3$. Then, $\mathcal{C}(N)$ is a 6-dimensional polytope, triangulated into 48 simplices corresponding to the 48 linear extensions (see Table 1):

$$\begin{aligned}
\sigma_1 &= \{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\}\\
\sigma_2 &= \{1\},\{2\},\{3\},\{1,2\},\{2,3\},\{1,3\}\\
\sigma_3 &= \{1\},\{2\},\{3\},\{1,3\},\{1,2\},\{2,3\}\\
\sigma_4 &= \{1\},\{2\},\{3\},\{1,3\},\{2,3\},\{1,2\}\\
\sigma_5 &= \{1\},\{2\},\{3\},\{2,3\},\{1,2\},\{1,3\}\\
&\vdots\ \ \vdots
\end{aligned}$$

According to (2), the vertices of R_{σ_1} are:

$(0,0,0,0,0,0)$, $(0,0,0,0,0,1)$, $(0,0,0,0,1,1)$, $(0,0,0,1,1,1)$, $(0,0,1,1,1,1)$, $(0,1,1,1,1,1)$, $(1,1,1,1,1,1)$

where the coordinates of a capacity μ in $\mathcal{C}(N)$ are given in the following order: ($\mu(\{1\})$,$\mu(\{2\})$,$\mu(\{3\})$,$\mu(\{1,2\})$, $\mu(\{1,3\})$,$\mu(\{2,3\})$). Hence, the centroid c_1 of this simplex is the barycenter of the 7 above vertices, which yields

$$c_1 = \Big(\frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}\Big).$$

Similarly, we obtain for R_{σ_2} the centroid c_2:

$$c_2 = \Big(\frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{6}{7}, \frac{5}{7}\Big).$$

After computing the centroids of $R_{\sigma_1}, \ldots, R_{\sigma_{48}}$, the centroid c of $\mathcal{C}(N)$ is the average of all these centroids:

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

Exact computation can be done also for $n = 4$, but not for greater values, as the number of linear extensions is too big. Note that the centroid, whose computation is based on linear extensions, inherits the same properties as the random variables $\boldsymbol{\mu}(S)$, i.e., denoting by $c(S)$ the coordinate of c pertaining to subset S, $c(S)$ depends only on the cardinality of S, and $c(N \setminus S) = 1 - c(S)$. This was already remarked by Miranda and Combarro [11]. The coordinates of the centroid for $n = 4$ are:

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

5.3 Experimental results

In order to do a fair comparison of the different methods, we must determine a suitable value of T (number of steps) for the Markov chain generator. Indeed, theoretically, if T tends to infinity, the obtained distribution tends to the exact distribution, but the computation time tends also to infinity. We propose to choose T so that the performance of the Markov chain method is approximately equal to the performance of the 2-layer method.

Determination of T in the Markov chain method. With $n \leqslant 4$, as the exact centroid is known, we propose to measure the performance using the distance (taken here as the sum of absolute values) between the theoretical centroid and the obtained centroid. Specifically, the procedure works as follows:

(i) Generate 100,000 capacities by the 2-layer approximation method and then compute the distance between the obtained centroid and the exact centroid, denoted by $d(2Layer)$;

(ii) Fix a threshold $\epsilon = 0.0001$, and an initial value $T = 1000$;

(iii) Compute the distance between the centroid obtained by the Markov Chain method and the exact centroid, denoted by $d_T(MC)$. If $d_T(MC) - d(2Layer) > \epsilon$, add 50 to T, and compute again $d_T(MC)$ with the new value of T, repeat the procedure until $d_T(MC) - d(2Layer) < \epsilon$.

In order to reduce variance, for each value of T, we compute five times $d_T(MC)$ and take the average value. Eventually, for $n = 4$, we obtained $T = 1170$.

When $n > 4$, another measure of performance must be taken, as the theoretical distribution (and therefore the theoretical centroid) cannot be obtained any more. Then we use the symmetry properties of the centroid, namely that $c(S)$ depends only on the cardinality of S. This leads to the following procedure:

(i) Generate 10,000 capacities by 2 layer approximation method and then compute the sum of the standard deviations of $c(S)$ when $|S|$ is constant, denoted by $sd(2Layer)$.

(ii) Fix a threshold $\epsilon = 0.0001$, and an initial $T = 8500$.

(iii) Compute the sum of the standard deviations of $c(S)$ when $|S|$ is constant using Markov Chain method, denoted by $sd_T(MC)$. If $sd_T(MC) - sd(2Layer) > \epsilon$, add 100 to T, and compute again $sd_T(MC)$ with new T, repeat the procedure until $sd_T(MC) - sd(2Layer) < \epsilon$.

As above, to reduce variance, for each value of T, we compute five times $sd_T(MC)$ and take the average value. Eventually, for $n = 5$, we obtained $T = 9000$.

Comparison of distributions We first perform a qualitative evaluation by examining the histograms of $\boldsymbol{\mu}(S)$ produced by the different methods.

Figures 4, 5, 6 and 7 show the histograms obtained for all subsets $S \in 2^N \setminus \{\varnothing, N\}$ for $n = 4$, respectively for the exact method, the 2-layer approximation method, the Markov chain generator [10], and the random node generator [9]. An inspection of the figures reveals that, qualitatively, all methods yield histograms with the required symmetry properties, as expected from Lemma 2. Moreover, all methods give histograms quite close to the exact distribution, except the random node generator, which yields distributions quite far from the exact ones. For this reason, we do not consider any more this method and focus on the comparison between the Markov chain method and the 2-layer approximation method.

Figures 8 and 9 show the histograms of $\boldsymbol{\mu}(S)$ obtained when $n = 5$ for the 2-layer approximation method and the Markov chain method, choosing one subset S of each cardinality 1, 2, 3 or 4. One can see that both method perform well, in terms of the expected symmetry between $\boldsymbol{\mu}(S)$ and $\boldsymbol{\mu}(N \setminus S)$.

We now perform a quantitative evaluation of the obtained distributions, using the Kullback-Leibler divergence method. Given two discrete probability distributions p, q on the same universe X, the Kullback-Leibler divergence is defined as

$$\mathbb{D}_{KL}(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}. \tag{14}$$

The smaller the value, the closer are the two distributions. As $\boldsymbol{\mu}(S)$ is a continuous distribution on $[0, 1]$, one must discretize it, dividing the interval $[0, 1]$ into segments of length δ. We choose $\delta = 0.01$ for our computation, a value much finer than the one used for the histograms in

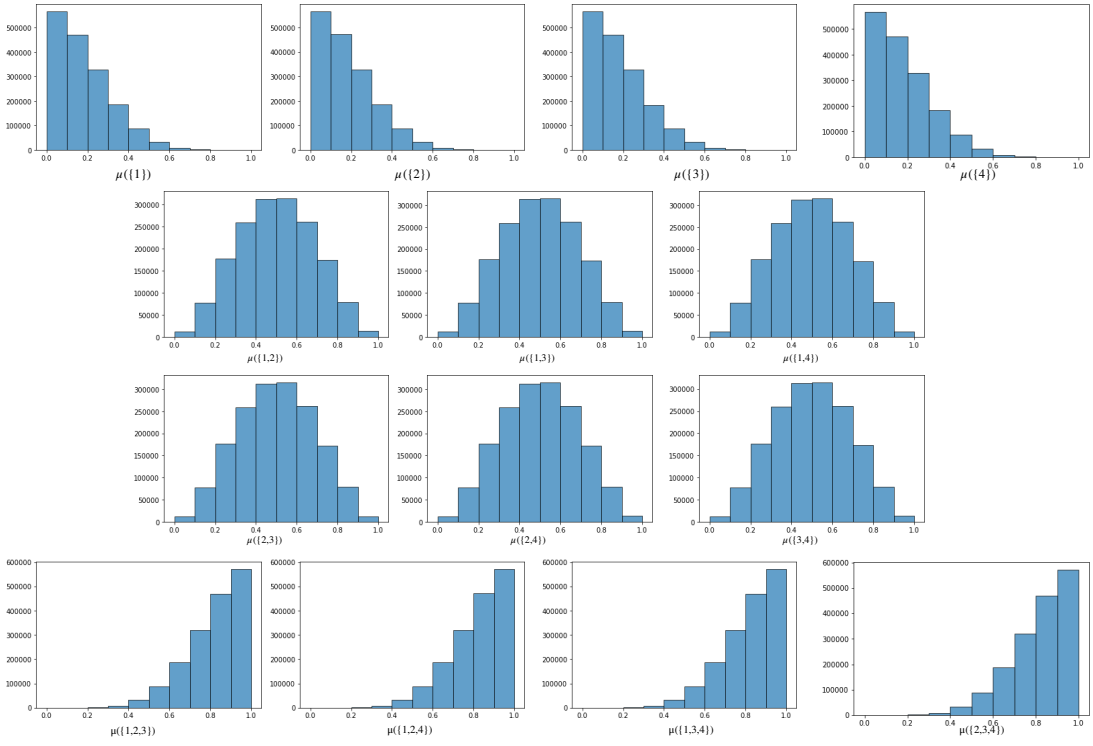


Figure 4: Histograms of $\boldsymbol{\mu}(S)$ for $n = 4$ and the exact method

figures 4 to 9 ($\delta = 0.1$). Let us denote by $\mu_{MC}(S)$ the discrete distribution of $\boldsymbol{\mu}(S)$ obtained by the Markov chain method, and by $\mu_{2L}(S)$ the one obtained by the 2-layer approximation method.

When $n = 4$ or smaller, as the exact distributions are known, we take q as the exact distribution in (14), and we compute the sum of the Kullback-Leibler divergence of $\mu_{MC}(S)$ and $\boldsymbol{\mu}(S)$, $\forall S \in 2^N$, denoted as $S^4_{KL}(\mu_{MC})$:

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

and

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

When $n > 4$, as exact distributions are no more available, we use the symmetry properties and compute the sum of the Kullback-Leibler divergence of $\mu_{MC}(S)$ and $\mu_{MC}(S')$, $\forall S, S' \in 2^N$ such that $|S| = |S'|$, denoted as $S^N_{KL}(\mu_{MC})$:

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

and

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

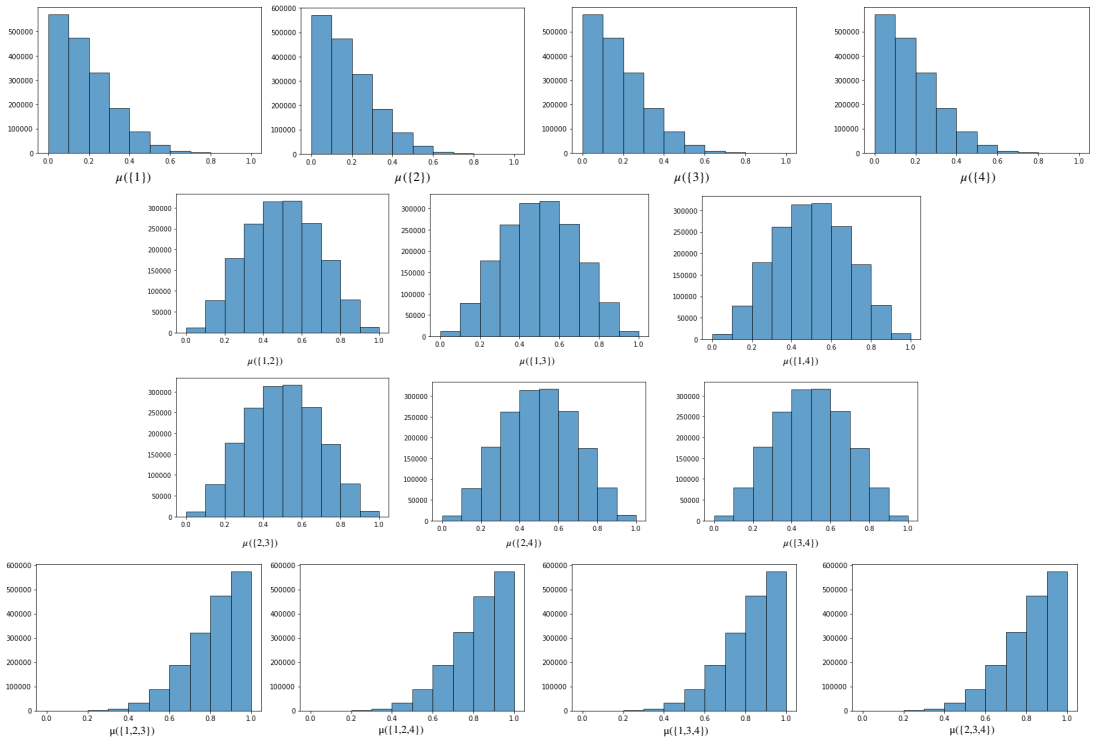


Figure 5: Histograms of $\boldsymbol{\mu}(S)$ for $n = 4$ and the 2-layer approximation method

$S^4_{KL}(\mu_{MC})$	$S^4_{KL}(\mu_{2L})$	$S^5_{KL}(\mu_{MC})$	$S^5_{KL}(\mu_{2L})$	$S^6_{KL}(\mu_{MC})$	$S^6_{KL}(\mu_{2L})$
0.061	0.059	2.41	2.24	8.24	8.41

Table 2: Kullback-Leibler divergences between the exact distribution and the distributions obtained by the 2-layer approximation method and the Markov chain method ($n = 4$), and Kullback-Leibler divergences testing the symmetry of distributions ($n = 5, 6$)

Table 2 shows the experimental result for $n = 4, 5$ and 6 (We take $T = 80,000$ for the Markov Chain Generator when $n = 6$). When $n = 4$, the result shows the divergence with the exact distribution (the smaller the value, the better the approximation). Both values are very close, indicating that the 2-layer approximation method yields distributions of equal quality with those obtained by the Markov chain method. When $n = 5$ or $n = 6$, the result shows to which extent the distributions are symmetric, i.e., depending only on the cardinality of S (here also, the smaller the value, the more symmetric are the distributions). We see that the values obtained by both methods are very close, without a clear advantage for one method or the other. Considering the Markov chain method as a benchmark, we can conclude that the 2-layer approximation method yields distributions of very similar quality to those obtained by the Markov chain method.

Computation time We have measured the computation time needed for generating 10,000 capacities by the Markov chain and the 2-layer approximation methods, using a 3.2 GHz PC with 16 GB of RAM. The results are given in Table 3, for $n = 4$, 5, 6, and 7. When $n \geqslant 6$, our procedure to obtain a suitable value of T becomes difficult to apply. Indeed, the huge

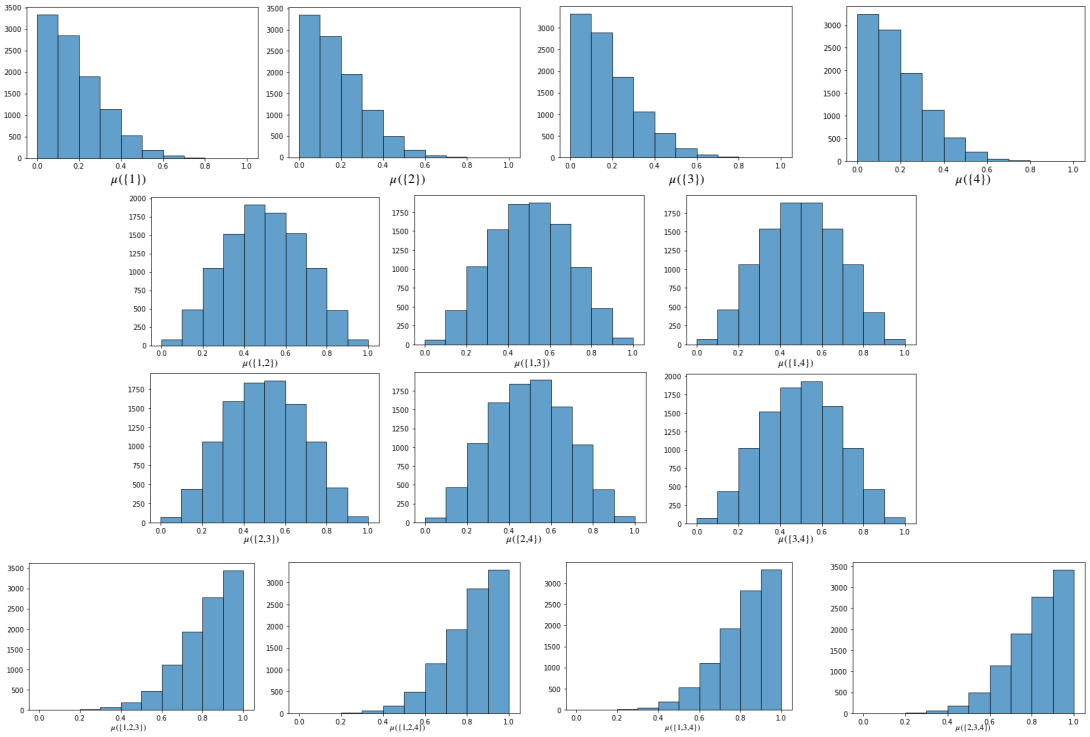


Figure 6: Histograms of $\boldsymbol{\mu}(S)$ for $n = 4$ and the Markov chain method

number of linear extensions obliges to significantly increase the number of sample capacities, and 10,000 generated capacities is no longer enough. As T is also dramatically increasing, the computational cost of the procedure becomes too large to find an exact value of T. Keeping a sample size of 10,000 capacities, we have found that, for $n = 6$, when $T = 80{,}000$ the difference between $sd_T(MC)$ and $sd_T(2Layer)$ is is around 0.0005, which is about 14% of $sd_T(2Layer)$; for $n = 7$, a similar difference is obtained when T =500,000.

As it can be seen, for an equal performance, the 2-layer approximation method is much faster than the Markov chain method.

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	≈ 1500	≈ 9000
	T	1170	9000	80,000	500,000

Table 3: Comparison of CPU time (s) for generating 10,000 capacities

5.4 Related literature

Several methods are used in the literature to measure the performance of a generator. Combarro et al. [6] use the centroid and its properties of symmetry mentioned above. They compute via an index they define if the coordinates of the average of the generated capacities are indeed depending only on the cardinality and if the symmetry S vs. $N \setminus S$ is verified. This, however, is not sufficient to ensure that the average capacity is close to the centroid.

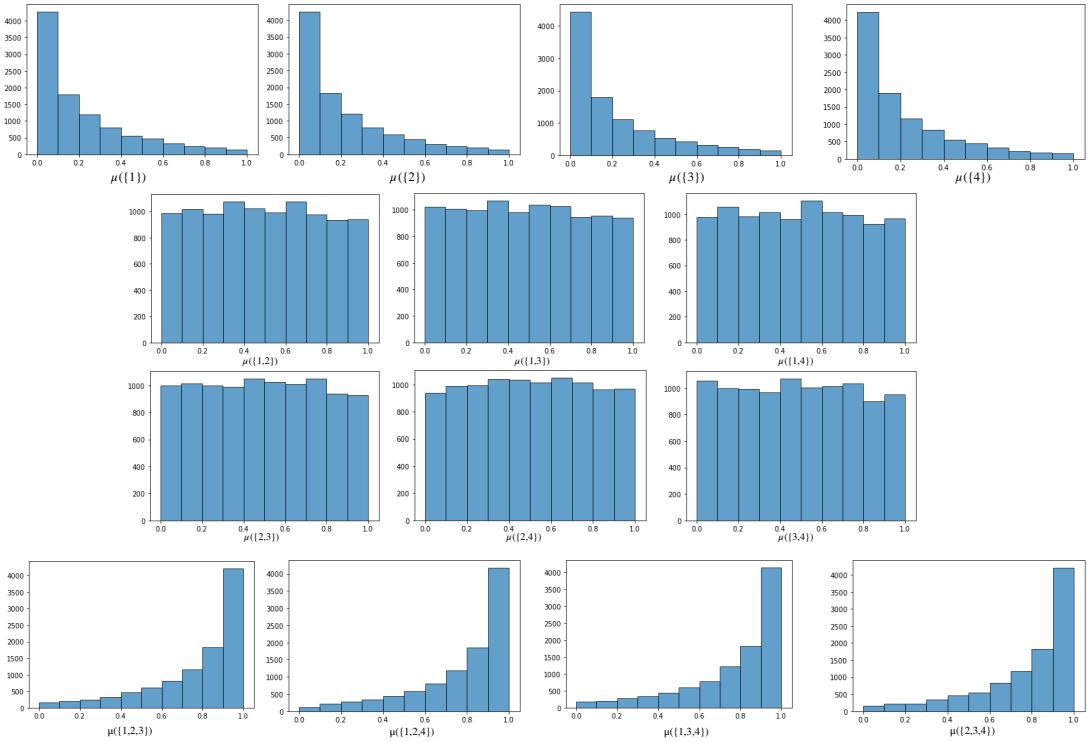


Figure 7: Histograms of $\boldsymbol{\mu}(S)$ for $n = 4$ and the Random Node Generator

In [7], Combarro et al. propose to measure the total variation distance to the uniform distribution, that is, to compute $\frac{1}{2}\sum_l |p_l - \frac{1}{e(p)}|$ with p_l the probability of obtaining linear extension l with a given capacity generator, and $e(p)$ is the total number of linear extensions.

Another method used in Havens and Pinar [9] is to compute the distribution of $w_{\sigma(i)}$ for $i \in N$ with $w_{\sigma(i)} = \mu(A_i) - \mu(A_{i-1})$ and $A_i = \{x_{\sigma(1)}, \ldots, x_{\sigma(i)}\}$, from histograms, and then to compute its entropy. The authors expect to have an entropy as large as possible, invoking the reason that a good random generator should provide a distribution as uniform as possible. However, the distribution of $w_{\sigma(i)}$ is impossible to get analytically and is far to be uniform. The only fact which is known is that these distributions, by the symmetry argument, do not depend on σ.

6 Competing interests

The authors have no relevant financial or non-financial interests to disclose.

7 Funding

The authors did not receive support from any organization for the submitted work.

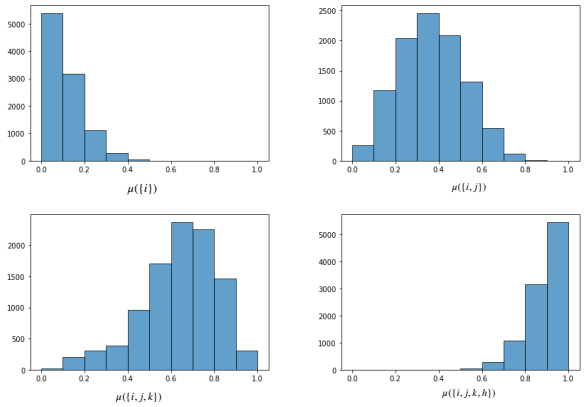


Figure 8: Histograms of $\boldsymbol{\mu}(S)$ for $n = 5$ and the 2-layer approximation method

References

[1] G. Beliakov. On random generation of supermodular capacities. *IEEE Tr. on Fuzzy Systems*, 30:293–296, 2022.

[2] G. Brightwell and P. Tetali. The number of linear extensions of the Boolean lattice. *Order*, 20:333–345, 2003.

[3] G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8:225–242, 1991.

[4] R. Bubley and M. Dyer. Faster random generation of linear extensions. *Discrete Mathematics*, 201:81–88, 1999.

[5] G. Choquet. Theory of capacities. *Annales de l'Institut Fourier*, 5:131–295, 1953.

[6] E. F. Combarro, I. Díaz, and P. Miranda. On random generation of fuzzy measures. *Fuzzy Sets and Systems*, 228:64–77, 2013.

[7] E. F. Combarro, J. Hurtado de Saracho, and I. Díaz. Minimals Plus: an improved algorithm for the random generation of linear extensions of partially ordered sets. *Information Sciences*, 501:50–67, 2019.

[8] M. Grabisch. *Set Functions, Games and Capacities in Decision Making*, volume 46 of *Theory and Decision Library C*. Springer, 2016.

[9] T. C. Havens and A. J. Pinar. Generating random fuzzy (capacity) measures for datafusion simulations. In *IEEE Symposium Series on Computational Intelligence (IEEE SSCI2017)*, pages 1–8, 2017.

[10] A. Karzanov and L. Khachiyan. On the conductance of order Markov chains. *Order*, 8:7–15, 1991.

[11] P. Miranda and E. Combarro. On the structure of some families of fuzzy measures. *IEEE Tr. on Fuzzy Systems*, 15(6):1068–1081, 2007.

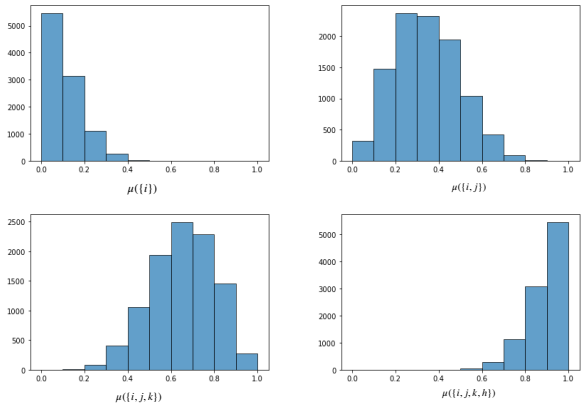


Figure 9: Histograms of $\boldsymbol{\mu}(S)$ for $n = 5$ and the Markov chain method

[12] P. Miranda and P. García-Segador. Bottom-up: a new algorithm to generate random linear extensions of a poset. *Order*, 36:437–462, 2019.

[13] P. Miranda and P. García-Segador. Applying Young diagrams to 2-symmetric fuzzy measures with an application to general fuzzy measures. *Fuzzy Sets and Systems*, 379:20–36, 2020.

[14] P. Miranda and P. García-Segador. Combinatorial structure of the polytope of 2-additive measures. *IEEE Transactions on Fuzzy Systems*, 28:2864–2874, 2020.

[15] R. Stanley. Two poset polytopes. *Discrete and Computational Geometry*, 1:9–23, 1986.

[16] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.

[17] T. Talvitie, T. Niinimäki, and M. Koivisto. The mixing of Markov chains on linear extensions in practice. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 524–530, Melbourne, Australia, 2017.

A Study of posets T_H closed under intersection

We give here a characterization of posets T_H closed under intersection, as well as the proof of Proposition 1.

We characterize now the form of any T_H closed under intersection. To this end, we need to distinguish two cases:

(i) Situation 1: for any node y of the lower layer, $|\mathsf{prec}(y)| \leq 2$.

(ii) Situation 2: there exists a unique y of the lower layer s.t. $|\mathsf{prec}(y)| = h$, with h the number of nodes in the upper layer, and $h \geqslant 3$.

We claim that under the assumption of closure under intersection, there is no other alternative.

Lemma 3. For any T_H closed under intersection there is no other possibility than Situations 1 and 2.

Proof. Let h be the number of nodes in the upper layer. We begin by remarking that if a node of the lower layer has h predecessors, it must be unique as it is the intersection of all nodes of the upper layer.

The cases where $h = 1, 2$ are trivial. For $h = 3$, either the number of predecessors of a lower node is at most 2, or it is 3, so either Situation 1 or 2 arises.

Assume then $h > 3$. Suppose there exist a node T at the lower level with $\mathsf{prec}(T)$ containing at least three nodes, say $T \cup \{1\}, T \cup \{2\}, T \cup \{3\}$ (therefore $T \subseteq N \setminus \{1, 2, 3\}$). Assume by contradiction that there exist S in the upper level such that $T \notin \mathsf{succ}(S)$. Since T_H is closed under intersection, there must exist a successor common to $T \cup \{1\}$ and S. As $T \not\subseteq S$, this successor has the form $(T \setminus \{i\}) \cup \{1\}$ for some $i \in T$. Therefore, $S \supseteq (T \setminus \{i\}) \cup \{1\}$. Similarly, $T \cup \{2\}$ and S must have a common successor, which implies that $2 \in S$. Similarly, we have also that $3 \in S$. Then

$$|S| \geq 3 + |T \setminus \{i\}| = |T| + 2,$$

which is impossible by definition of T_H. □

Proposition 5. Assume T_H is closed under intersection, let $N' := \bigcup_{x \in \text{upper layer}} x$, with $|N'| = n' \leqslant n$, and denote by h the number of nodes in the upper layer, of cardinality ℓ.

The two situations are characterized as follows:

- Situation 1:
  - Case 1: $h = 1$, i.e., there is only one element in the upper layer;
  - Case 2: $h \geqslant 2$. Then $\ell = n' - 1$, i.e., the nodes of the upper layer are of the form $N' \setminus \{i\}$. We have $\frac{h(h-1)}{2}$ nodes in the lower layer having exactly 2 predecessors.
- Situation 2: $h = n' - \ell + 1$ and $\ell < n' - 1$. Nodes of the upper level are of the form $S \cup \{i\}$, and the lower layer contains the node S with h predecessors, and possibly other nodes with 0 or 1 predecessor.

Proof. For Situation 1: Suppose we have h nodes in the upper layer, $h \geqslant 2$. As T_H is closed under intersection, any two nodes in the upper layer have a common successor in the lower layer. For x, x' in the upper layer, there is one and exactly one element in the lower layer successor of both nodes. Thus, there are exactly $\frac{h(h-1)}{2}$ elements in the lower layer having 2 predecessors.

Hence the $\frac{h(h-1)}{2}$ pairs of nodes x, x' in the upper layer yield $\frac{h(h-1)}{2}$ distinct elements in the lower layer. Indeed, if they were not distinct, some node in the lower level would have at least 3 predecessors.

It remains to prove that $\ell = n' - 1$. To this aim, let us take two nodes in the upper layer with common successor the set $S \subseteq N'$. Then these two nodes must be of the form $S \cup \{i\}$ and $S \cup \{j\}$. If $h = 2$, we have $N' = S \cup \{i, j\}$, which proves the assertion. Suppose now that $h \geqslant 3$ and consider in the upper layer a set $T \neq S \cup \{i\}, S \cup \{j\}$. We claim that $N' = S \cup \{i, j\}$, i.e., $T \subseteq S \cup \{i, j\}$, which proves our assertion. Suppose by contradiction that T contains an element l such that $l \notin S \cup \{i, j\}$. Let us write $T = S' \cup \{l\}$ with $S' \not\ni l$. As T_H is closed under intersection, T and $S \cup \{i\}$ have a common successor, which must be S', because l does not belong to the successor and the successor must have $\ell - 1$ elements. Similarly, the common successor of T and $S \cup \{j\}$ must be S'. But then S' has three predecessors, a contradiction.

For Situation 2: Denoting by $S_1, \ldots, S_h$ the nodes in the upper layer, we know by definition of Situation 2 that there exists S in the lower layer s.t. $S = S_1 \cap \cdots \cap S_h$. Hence, the nodes in

the upper layer have the form $S \cup \{i\}$ with $i \in N' \setminus S$. Consequently, $h = n' - |S| = n' - \ell + 1$, i.e., $\ell = n' - h + 1 < n' - 1$ since $h \geqslant 3$.

Moreover, any two nodes $S \cup \{i\}$ and $S \cup \{j\}$ of the upper layer have a unique common successor, which is S. Hence, any $T \neq S$ in the lower layer must have 0 or 1 predecessor. □

Of course, symmetric results can be established for the two bottom layers B_H.

Figures 10 and 11 illustrate Situations 1 and 2, as well as the function n_x.

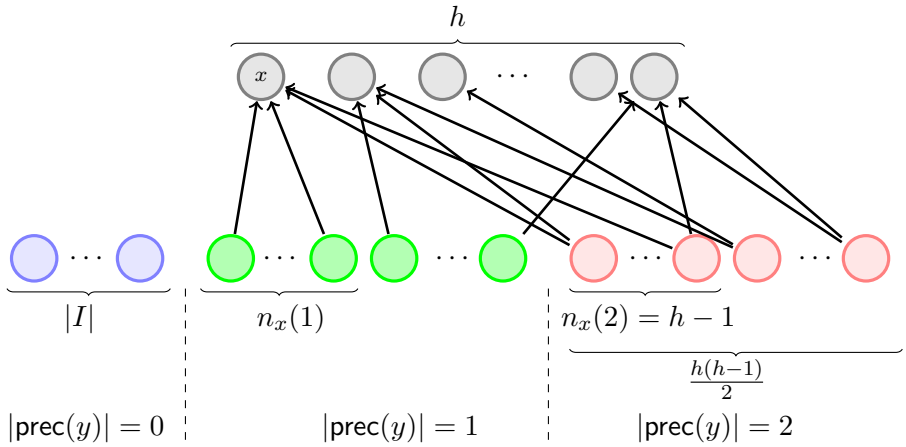


Figure 10: Situation 1.

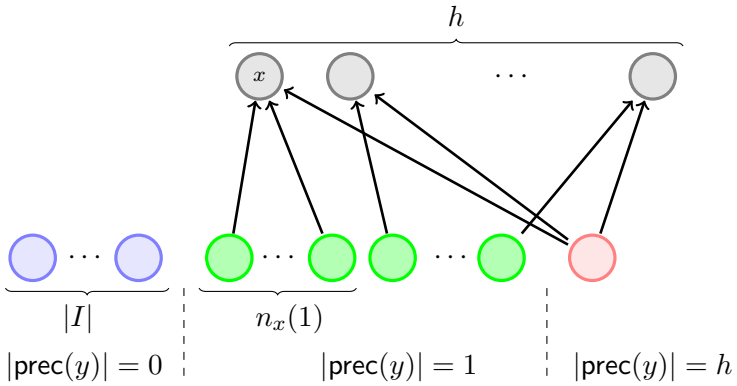


Figure 11: Situation 2.

Proof of Proposition 1.

(i) Observe that $n_x(0) = k - \mathsf{succ}(x)$, where k is the number of nodes in the lower layer. As n_x is invariant, $\mathsf{succ}(x)$ does not depend on x.

(ii) We distinguish two cases: Situation 1 and Situation 2.

1. Suppose we are in Situation 1, i.e., $|\mathsf{prec}(y)| = 0$, 1 or 2 for every y in the lower layer. We have by definition $n_x(0) = k - |\mathsf{succ}(x)|$. As $|\mathsf{succ}(x)|$ is constant with x by blancedness, $n_x(0)$ does not depend on x. Now, as $T_H[h, k, 0]$ is closed under intersection,

a given node x has $h-1$ distinct successors y given by $\mathsf{succ}(x) \cap \mathsf{succ}(x')$ for each $x' \neq x$ in the upper layer. Therefore $|\mathsf{prec}(y)| = 2$ for each of them, so that $n_x(2) = h - 1$. Finally, $n_x(1) = k - n_x(2) - n_x(0)$, which establishes the result.

2. Suppose we are in Situation 2. From Proposition 5, we know that $|\mathsf{prec}(y)| = 0$, 1 or h. It suffices to prove that n_x is invariant w.r.t. x. We know from Proposition 5 again that there is a unique node y with $|\mathsf{prec}(y)| = h$, so that clearly $n_x(h) = 1$ for every x. Similarly as above, $n_x(0) = k - |\mathsf{succ}(x)|$, which is invariant w.r.t. x. Finally, $n_x(1) = k - n_x(h) - n_x(0)$, which establishes the result.

(iii) Take $T_H = \{123, 124, 23, 24\}$. Then n_x is invariant but T_H is not closed under intersection.

(iv) Take $H = 2^N \setminus \{N, \varnothing\}$, with $|N| = n$. Then $f_x(y) = 2$ for every y successor of x, so that clearly n_x is invariant.

(v) Take any two nodes x, x' of the upper layer. We have to prove that $|\mathsf{succ}(x) \cap \mathsf{succ}(x')| = 1$. The property $|x''| = n - 1 = |\mathsf{succ}(x'')|$ for evey node x'' of the upper layer implies that all subsets of x'' are successors of x''. As $n \geqslant 3$, successors of x'' are nonempty sets. In particular $x \cap x'$ is a nonempty set successor of both x and x', hence $|\mathsf{succ}(x) \cap \mathsf{succ}(x')| = 1$, so that T_H is closed under intersection. In addition, $|\mathsf{succ}(x'')| = n - 1$ for evey node x'' of the upper layer implies that T_H is balanced. It follows from (ii) that T_H is regular.