On random generation of fuzzy measures

E. F. Combarro[a], I. Díaz[b], P. Miranda[c]

[a]*University of Oviedo*
Campus de Viesques s/n, 33204, Gijón (Spain)
elias@aic.uniovi.es
[b]*University of Oviedo*
Campus de Viesques s/n, 33204, Gijón (Spain)
sirene@uniovi.es
[c]*Complutense University of Madrid*
Plaza de Ciencias, 3, 28040 Madrid (Spain)
pmiranda@mat.ucm.es

Abstract

In this paper we deal with the problem of obtaining a random procedure for generating fuzzy measures. We use the fact that the polytope of fuzzy measures is an *order polytope*, so that it has special properties that allow to build a uniform algorithm. The procedure that we derive also applies to any subfamily of fuzzy measures that is an order polytope. We study the applicability of this algorithm to the polytope of fuzzy measures, showing that the complexity grows dramatically with the cardinality of the referential set. Then, we propose another heuristic that reduces the complexity and might be applied instead. We finish comparing the performance of the proposed heuristic with other procedures appearing in the literature.

1. Introduction

Consider a finite referential set of n elements, $X = \{1,\ldots,n\}$ and let us denote by $A,B,\ldots$ the subsets of X. The set of subsets of X is denoted by $\mathscr{P}(X)$.

A **fuzzy measure** [31] (or **capacity** [5] or **non-additive measure** [10]) over X, is a set function $\mu : \mathscr{P}(X) \to [0,1]$ satisfying

- $\mu(\emptyset) = 0, \mu(X) = 1,$
- $\mu(A) \leq \mu(B)$ for all $A,B \in \mathscr{P}(X)$ such that $A \subseteq B$.

Fuzzy measures have been applied to many different fields, such as Multicriteria Decision Making, Decision Under Uncertainty and Risk, Game Theory, Welfare Theory or Combinatorics (see [13] for a review of theoretical and practical applications of fuzzy measures). Moreover, they are included in the field of Aggregation Operators, that constitutes a major research topic nowadays [12].

An interesting problem arising in the practical use of fuzzy measures is the identification of the fuzzy measure modelling a concrete situation. In this case, we have to deal with the fact that the number of coefficients needed to define a fuzzy measure is $2^n - 2$ for a referential of cardinality n; thus, the complexity grows exponentially. The problem of identification (sometimes restricted to a subfamily of fuzzy measures) has attracted the attention of many researchers, and many procedures to solve this problem have been proposed (see for example [34], [19], [32], [14], [1], [27], [6]). The information background used by each method also varies; sometimes, a sample data is supposed to be available; some methods use a questionnaire to get information, ...; the data considered in the procedure can be numerical or just ordinal; and so on.

Once an algorithm is suggested, it is necessary to test whether it works properly or not. Usually, a fuzzy measure is considered, some data are generated (possibly with some random noise), and the corresponding procedure is applied. If the procedure works properly, it should obtain a fuzzy measure *near* the considered measure. However, in order to properly evaluate the performance of a procedure, it should be tested not only in a particular case but in many different situations, and the fuzzy measure considered in each case should be chosen randomly.

Interestingly enough, the problem of generating fuzzy measures in a random way can also be straightforward applied in the problem of identification of fuzzy measures. In [6], we have proposed a method based on genetic algorithms to deal with the problem of identification of some convex families of fuzzy measures; in that paper, the cross-over operator considered is the convex combination. This algorithm is very fast and the simulations carried out suggest that it is stable with respect to the presence of noise. However, the convex combination reduces the search region in each iteration. To bear on this problem, the only option is to use as initial population the set of vertices of the corresponding subfamily (see [6] for details). On the other hand, the number of extreme points of the set of fuzzy measures is very large (see Table 2 below) and this makes the use of the set of vertices unfeasible for large values of n (and $n = 8$ is already large!). Consequently, we are dragged to seek another initial population and a possibility is to initialize the population with a uniform generator of fuzzy measures.

To our knowledge, there is not a method in the literature to randomly generate a fuzzy measure. This is the problem that we study in this paper.

We will denote by $\mathscr{FM}(X)$ the set of fuzzy measures on X. Identifying a fuzzy measure with the vector of its values on $\mathscr{P}(X)\setminus\{\emptyset, X\}$, it can be seen from the definition of fuzzy measure that $\mathscr{FM}(X)$ is a bounded polyhedron (i.e. a polytope) of $\mathbb{R}^{2^n-2}$. Consequently, the problem reduces to derive a procedure for the uniform random generation in a polytope. However, this is a complex problem in general and several methods have been suggested to cope with it. Indeed, the uniform random generation in a polytope is a hot problem in Computer Sciences nowadays (see for example [11] and [29]).

In our case, we will use the fact that $\mathscr{FM}(X)$ belongs to a special family of polytopes, the so-called *order polytopes*. Then, as we will explain below, the problem simplifies and it is possible to obtain a uniform procedure.

However, we will show that the complexity of this procedure increases very quickly when $|X|$ grows. In order to cope with this problem, we propose an heuristic that can be applied to any order polytope, although they may lead to non-uniform procedures.

The rest of paper is organized as follows. As the notion of order polytope is central in our approach and in order to be self-contained, we will begin with the basic notions and properties about order polytopes that are needed in the paper. In Section 3, we provide a uniform algorithm for the random generation of fuzzy measures and we study its applicability. Next, in Section 4, we propose a new procedure to deal with the problem. Section 5 presents briefly some other possibilities appearing in the literature. Section 6 compares the behavior of all these procedures for the polytope $\mathscr{FM}(X)$. We finish with the conclusions and open problems.

2. Order polytopes and its random generation

Consider a finite poset $(P,\preceq)$ (or P for short) of p elements. We will denote the subsets of P by capital letters $A,B,\ldots$; elements of P are denoted a,b, and so on. If A is a subset of P, it inherits a structure of poset from the restriction of $\preceq$ to A. In this case, we say that A is a **subposet** of P. A subposet $(A,\preceq)$ is a **chain** if for any $a,b\in A$, either $a\preceq b$ or $b\preceq a$, i.e.$\preceq$ is a total order.

A poset can be represented by the so-called *Hasse diagram*, a graph where $a\preceq b$ if and only if there is a sequence of connected lines upwards from a to b. An example of Hasse diagram is given in Figure 1.

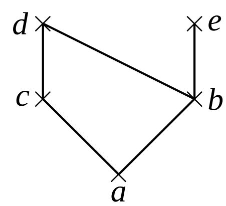

Figure 1: Example of Hasse diagram of a poset.

A subset I of P is an **ideal** or **downset**[1] if for any $a\in I$ and any $b\in P$ such that $b\preceq a$, it follows that $b\in I$. We will denote ideals by $I_1,I_2,\ldots$ Notice that with this definition the empty set is an ideal. The dual notion of an ideal is a **filter** or **upset**, i.e., a set that contains all upper bounds of its elements. We will denote by $\mathscr{I}(P)$ the set of all ideals of poset P.

Given two ideals I_1 and I_2 of P, we can define $I_1\cup I_2$ and $I_1\cap I_2$ as the usual union and intersection of subsets. It is trivial to check that $I_1\cup I_2$ and $I_1\cap I_2$ are ideals, too. In fact, $\mathscr{I}(P)$ is a lattice under set inclusion called the **lattice of ideals** of P (see [2]). The lattice of ideals of the previous poset is given in Figure 2.

An **extension** $(P,\preceq')$ of $(P,\preceq)$ is another poset over the same referential P such that $\preceq'$ is order-preserving (i.e. if $x\preceq y$, then $x\preceq' y$). A **linear extension** is an extension that is a chain. The linear extensions of the poset of Figure 1 are given in Figure 3.

Let us now turn to order polytopes. Given a poset $(P,\preceq)$, it is possible to associate to P, in a natural way, a polytope $O(P)$ over $\mathbb{R}^p$, called the **order polytope** of P (cf. [30]). The polytope $O(P)$ is formed by the p-uples f of real numbers indexed by the elements of P satisfying

- $0\leq f(a)\leq 1$ for every a in P,
- $f(a)\leq f(b)$ whenever $a\preceq b$ in P.

Thus, the polytope $O(P)$ consists of (the p-uples of images of) the order-preserving functions from P to $[0,1]$. It is a well-known fact [30] that $O(P)$ is a $0/1$-polytope, i.e. its extreme points are all in

[1] It should be noticed that the notions of downset and ideal are equivalent in posets but not in lattices. Namely, the lattice definition of ideal is stronger than that of downset, as it also requires that for any two elements in the ideal, the upper bound is also in the ideal (see [16]).

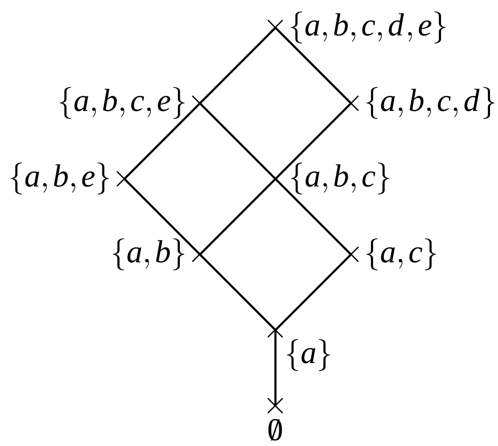

Figure 2: Lattice of ideals corresponding to poset of Figure 1.

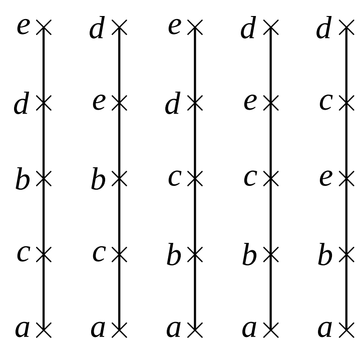

Figure 3: The set of linear extensions of the poset in Figure 1.

$\{0,1\}^p$. In fact, it is easy to see that the extreme points of $O(P)$ are exactly (the characteristic functions of) the filters of P. In this sense, the extreme point whose value is 1 for any element of P is identified with the filter P, while the extreme point whose value is 0 for any element of P is identified with the filter $\emptyset$. Applied to distributive lattices, the notion of order polytope has been also defined in [18] with the name of **geometric realization**.

It can be easily seen that the polytope $\mathscr{FM}(X)$ is the order polytope of the poset $(P,\preceq)$ where $P=\mathscr{P}(X)\backslash\{X,\emptyset\}$ and $\preceq$ is the inclusion between subsets [7]. Thus, the problem of random generation of fuzzy measures reduces to obtain a random procedure for generating points in an order polytope.

Finally, let us treat the problem of random generation of order polytopes. There are several procedures to generate random points in a polytope; for example, we have the grid method [11], the sweep-plane method [22] and triangulation methods [11]. For order polytopes, we think that the triangulation methods are the best option because of the properties that order polytopes share; more details will become apparent below.

Consider $n+1$ affine independent points in $\mathbf{R}^m, m \geq n$, i.e. $n+1$ points of $\mathbf{R}^m$ in general position. The convex hull of these points is called a **simplex**. This notion is a generalization of the notion of triangle for the m-dimensional space.

The random generation in simplices is very simple and fast [28]. Indeed, if $x_1,\ldots,x_{n+1}$ are the vertices of the simplex, it suffices to generate random values $\alpha_1,\ldots,\alpha_{n+1}\in[0,1]$ such that $\sum_{i=1}^{n+1}\alpha_i=1$; the point generated is $\sum_{i=1}^{n+1}\alpha_i x_i$.

The *triangulation method* is based on the decomposition of the polytope into simplices; once the decomposition obtained, we assign to each simplex a probability proportional to its volume; next, these probabilities are used for selecting one of the simplices; finally, a random m-uple in the simplex is generated.

However, in general it is not easy to split a polytope in simplices. Moreover, even if we are able to decompose the polytope in a suitable way, we have to deal with the problem of determining the volume of each simplex in order to properly select one of them. This is the *Achilles heel* of the triangulation method and the reason for which its practical application is limited for general polytopes [29]. However, when dealing with order polytopes, the following result can be applied (see [25], pag. 304):

Theorem 1. *Let $(P,\preceq)$ be a poset of p elements.*

- *If $\preceq$ is a total order on P, then the corresponding order polytope is a simplex of volume $\frac{1}{p!}$.*
- *For any partial ordering $\preceq$ on P, the simplices of the order polytope of $(P,\leq)$, where $\leq$ is a linear extension of $\preceq$, cover the order polytope of $(P,\preceq)$ and have disjoint interiors. Consequently, $vol(O(P,\preceq))=\frac{1}{p!}e(\preceq)$, where $e(\preceq)$ denotes the number of linear extensions.*

These results are also outlined in [30].

Consequently, it suffices to generate randomly a linear extension of $\preceq$ and then generate a point in the corresponding simplex.

3. An exact procedure for the random generation of fuzzy measures

As explained in Section 2, $\mathscr{FM}(X)$ can be seen as an order polytope on $\mathbb{R}^{2^n-2}$, being n the number of elements of X. Therefore, the problem of randomly generating fuzzy measures corresponds to a random procedure for generating points in an order polytope and, as seen in the previous section, this reduces to produce linear extensions of the corresponding poset in a random way.

A method to generate linear extensions of a poset in a random way appears in [23]. This method is based on graph counting operations on the lattice of ideals representation of the given poset, and uses the fact that a linear extension can be identified with a path from the source (the empty ideal) to the sink (the whole poset). Then, once the lattice of ideals is built (an algorithm for generating the lattice of ideals of a poset appears in [24]), it suffices to select a path from the source to the sink in a random way.

Suppose we are at step i. Then, we are in an ideal $I \in \mathscr{I}(P)$ of the lattice of ideals and suppose w.l.g. $I := \{x_1, \ldots, x_{k-1}\}$. We have to move from I to another ideal $I \cup \{x_r\}$ with $r \geq k$. In order to select the element x_r we have to compute the number of paths from I to the sink such that next ideal considered is $I \cup \{x_r\}$ for each $r \geq k$. It can be seen that these numbers are given by the number of linear extensions of the filter $F := P \backslash (I \cup \{x_k\})$. Consequently, the probability of selecting an element x_r is given by

$$\frac{LEF(P \backslash (I \cup \{x_k\}))}{LEF(P \backslash I)}$$

where $LEF(A)$ denotes the number of linear extensions of filter A.

The use of the lattice of ideals instead of directly enumerating all linear extensions is justified by the fact that the number of linear extensions is in general much larger than the number of ideals (see Table 1 obtained in [23]).

$\|P\|$	$\|\Omega(P)\|$	$\|\mathscr{I}(P)\|$
5	6	9
10	5.4×10^2	33
15	2.39×10^5	148
20	1.13×10^9	518
25	1.07×10^{12}	953
30	7.83×10^{14}	1406
35	5.57×10^{17}	2637

Table 1: The number of ideals ($|\mathscr{I}(P)|$) compared to the number of linear extensions ($|\Omega(P)|$) in a poset

The posets considered in the table have been obtained by choosing uniformly at random $|P|$ points out of a two-dimensional grid of size 20 by 20, equipped with the usual ordering. Of course, for other polytopes these figures could vary, but it can be seen that in general the number of linear extensions grows much faster than the number of ideals. In fact, the problem of counting the number of linear extensions of a poset is a ♯P-problem [3] (see [33] for the definition of ♯P-problem).

In [23] it is proved that this procedure provides a uniform algorithm for the random generation of points in an order polytope and consequently, it provides a uniform procedure for the random generation of fuzzy measures.

However, it cannot be applied in practice for large values of n (and $n = 8$ is already large). The reason lays in the fact that it is necessary to build the corresponding lattice of ideals and the number of ideals increases dramatically with n. To see this, note that a filter F is characterized by its minimal elements (*a* is minimal in F if $b \not\preceq a, \forall b \in F$) because F can be written as $F = \{b \in P : \exists a \in F | a \preceq b\}$; the minimal elements of F determine an antichain in P. Thus, the number of filters (and consequently the number of ideals) of $\mathscr{P}(X) \backslash \{\emptyset, X\}$ coincides with the cardinality of the set of antichains of $\mathscr{P}(X) \backslash \{\emptyset, X\}$, and it has been proved in [6] that this number coincides with the sequence of Dedekind numbers [9], whose general form is a long-standing open problem in Combinatorics. The first Dedekind numbers are given in Table 2.

n	Dedekind numbers
1	1
2	4
3	18
4	166
5	7,579
6	7,828,352
7	2,414,682,040,996
8	56,130,437,228,687,557,907,786

Table 2: Number of vertices of $\mathscr{FM}(X)$

From the values in this table, we conclude that we cannot use the lattice of ideals when n is large in order to build a procedure for random generation on $\mathscr{FM}(X)$, due to the huge number of elements that need to be stored.

On the other hand, this method is interesting for us because it serves to compare the performance of the different heuristics for small cardinalities. This method will be called *DeLoof* in this work.

4. A heuristic based on minimal elements

We propose the following method for generating linear extensions. Consider $(P, \preceq)$. In a first step, we look for the elements of P that can be ranged in the first place in the linear extension. We will call these elements of the poset the *minimal elements* of P, i.e. we start considering the set

$$\{a \in P \,|\, x \not\prec a, \forall x \in P\}.$$

We choose randomly an element in this set, say a_1. This element is ranged in the first position in the linear extension; next, we remove a_1 from P, so that we obtain a subposet of P, that we will denote P_2.

At iteration i, a minimal element of poset P_i is selected randomly, say a_i. This element will be in the i-th position in the linear extension. Then, a_i is removed from P_i and we consider the new poset P_{i+1}. We continue this process until all elements of P are considered. This method will be called *Minimals* in this work. Notice that this is a simplified version of the exact method explained in the previous section, but with no need to compute the ideals or their probabilities, so that it can be applied for big cardinalities.

This procedure does not provide a uniform method for general posets, as the following example shows:

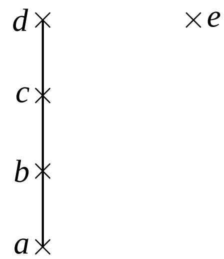

Figure 4: Example of Hasse diagram of a poset.

Example 1. *Consider the poset whose Hasse diagram is given in Figure 4.*

In this case, we have five possible linear extensions, but simple calculi show that this procedure, instead of generating each linear extension with probability $\frac{1}{5}$, *assigns the following probabilities:*

Linear extension	*Probability*
$e-a-b-c-d$	$\frac{1}{2}$
$a-e-b-c-d$	$\frac{1}{4}$
$a-b-e-c-d$	$\frac{1}{8}$
$a-b-c-e-d$	$\frac{1}{16}$
$a-b-c-d-e$	$\frac{1}{16}$

Thus, Minimals provides a distribution on the set of linear extensions that is far away from uniformity.

However, remark that the poset considered in the previous example is very asymmetrical. Then, this procedure might be a suitable method for $\mathscr{FM}(X)$ due to the symmetrical structure of the poset $\mathscr{P}(X)\backslash\{\emptyset, X\}$ (see [8]). To illustrate this situation, the poset $\mathscr{P}(X)\setminus\{\emptyset, X\}$ when $X = \{1,2,3\}$ is depicted in Figure 5.

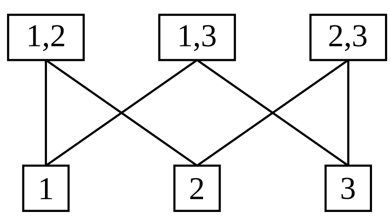

Figure 5: Poset $\mathscr{P}(X)\setminus\{\emptyset, X\}$ when $X = \{1,2,3\}$.

On the other hand, this method can be further modified to select the elements according to an estimated probability instead of choosing them randomly. This method will be called *Minimals*$_p$. In this case, if $\{e_1, e_2, \ldots, e_m\}$ is the set of minimal elements in the poset and we denote by n_i the number of elements in the poset greater than or equal to e_i, then the probability of selecting e_i as next element in the linear extension is estimated by

$$\frac{n_i}{\sum_{i=1}^{m} n_i}.$$

Let us show an example of this procedure.

Example 2. *Consider again the poset whose Hasse diagram is given in Figure 4. The procedure* Minimals$_p$ *assigns the following probabilities:*

Linear extension	*Probability*
$e-a-b-c-d$	$\frac{1}{5}$
$a-e-b-c-d$	$\frac{4}{5}\times\frac{1}{4}=\frac{1}{5}$
$a-b-e-c-d$	$\frac{4}{5}\times\frac{3}{4}\times\frac{1}{3}=\frac{1}{5}$
$a-b-c-e-d$	$\frac{4}{5}\times\frac{3}{4}\times\frac{2}{3}\times\frac{1}{2}=\frac{1}{5}$
$a-b-c-d-e$	$\frac{4}{5}\times\frac{3}{4}\times\frac{2}{3}\times\frac{1}{2}=\frac{1}{5}$

Thus, Minimals$_p$ *provides a more uniform distribution than* Minimals. *It is easy to check that Minimals$_p$ also provides uniform sampling for the linear extensions of the posets in Figures 1 and 5.*

5. Other heuristics for random generation of linear extensions

In this section we present two other heuristics proposed in the literature to approximate the problem of random generation of linear extensions.

5.1. Lerche and Sörensen algorithms

This approach is based on deciding the order between pairs of incomparable elements. The method is developed in [20] and its main steps are listed below.

1. Consider all the incomparable pairs in the poset.
2. Select one of them randomly.
3. Decide at random which element of the pair will be the lower in the linear extension.
4. Compute the transitive closure according to the new order constraint introduced.
5. Repeat the process until all incomparable pairs are included into the linear extension.

Note that in the third step the order between two incomparable elements is decided. The first approach to do that is to assume that the order between two incomparable pair of objects is uniformly chosen (we call this method *Lerche*). However, the same problems arising in *Minimals* appear in this case.

In fact, in order to generate a linear extension in a random way, it is necessary to obtain previously for each pair of incomparable elements the probability of the first element of the pair is before the second element of the pair in a linear extension. This is known as the *Mutual Ranking Probability*. However, this Mutual Ranking Probability is sometimes impossible to compute exactly since all linear extensions would be needed.

Instead, in [21] it is proposed to estimate these values by considering the number of elements above and below each element of the pair. Note that the maximum ranking position for an element is the total number of elements in the poset minus the number of elements above; similarly, the minimum ranking position an element can attain is the number of objects below this element plus one. Given a pair of incomparable elements and the maximum and minimum ranking positions of the elements of the pair, we can obtain the possible ranking positions in a random linear extension; finally, the probability of the first element of the pair is considered below the second element of the pair is estimated as the proportion of possible ranking positions in this condition. This method will be hereinafter called $Lerche_p$. The next example illustrates this step.

Example 3. *Consider the poset whose Hasse diagram is given in Figure 4 and let us apply the* $Lerche_p$ *algorithm to this example. First, all the incomparable pairs must be considered. In this example the incomparable pairs are* (a,e), (b,e), (c,e), (d,e). *Now one of them is randomly selected, say* (a,e).

Table 3: Number of Elements above, below, maximum and minimum ranking for *a* and *b*

Element	a	e
Number of elements above	3	0
Number of elements below	0	0
Maximum Ranking	2	5
Minimum Ranking	1	1

Therefore element a has 2 *possible ranking positions (positions* 1 *and* 2*) while element e can be placed in any position. In total there are* 8 *possible combinations since the situation where a and e share the same ranking are excluded. As* $a < e$ *is obtained in* 7 *cases, this gives an estimated mutual rank probability for* $a < e$ *of* 0.875*. Note that the real probability is* 0.80*.*

5.2. Based on Markov series

This procedure was firs proposed by Karzanov and Khachiyan in [17] (see also [15]).

Consider a poset $(P,\preceq)$ of p elements and a total order $(P,\leq)$ compatible with $\preceq$, i.e. a linear extension of this poset. Then, we can write this total order as a p-dimensional vector $\vec{x} := (a_1,\ldots,a_p)$ defined by $a_i \leq a_j \Leftrightarrow i \leq j$. Let us denote by $\sigma(i,j)\vec{x}$ the total order in which the elements in positions i and j has been interchanged, i.e. if $\vec{x} \equiv (a_1,\ldots,a_p)$, then

$$\sigma(i,j)\vec{x} \equiv (a_1,\ldots,a_{i-1},a_j,a_{i+1},\ldots,a_{j-1},a_i,a_{j+1},\ldots,a_p).$$

At the beginning, we consider a linear extension $\vec{x}_1$ of the original poset P. Then, we build a Markov chain starting from $\vec{x}_1$. Given the t-th element of the chain, element $\vec{x}_{t+1}$ is built as follows:

- Choose a number i in $\{1,\ldots,p-1\}$ according to the probability distribution

$$\phi(i) = \frac{6i(p-i)}{p^3-p},$$

and $c_0 \in \{0,1\}$ uniformly at random.

- If $c = 0$ or $\sigma(i, i+1)\vec{x}_t$ is not a linear extension of P, then $\vec{x}_{t+1} = \vec{x}_t$. Otherwise, $\vec{x}_{t+1} = \sigma(i, i+1)\vec{x}_t$.

In [4] it is shown that this evolves in limit to an uniform linear extension of $(P, \preceq)$, no matter the initial linear extension. See [4] and references in it for further details.

The problem with this procedure is that we cannot be sure whether we are near the limit or not, so that we do not know the number of iterations that should be done in order to be near uniformity. Besides, this is a procedure that remains valid for *any* poset, but it is not using the special properties of $\mathscr{P}(X)\backslash\{X, \emptyset\}$. We will denote this method by M_r, where r is teh number of iterations considered.

Wilson proved in [35] that the above algorithm also can be used if ϕ is a uniform distribution (as proposed originally by Karzanov and Khachiyan in [17]). This method will be denoted by M_r^k.

6. Simulations

This section is devoted to compare the performance of the different procedures for the random generation of fuzzy measures presented in the previous sections. We have used Java implementations of the algorithms described above and have conducted the experiments in a 2.5GHz PC with 4GB of RAM. To carry on this study each algorithm generates M linear extensions ($M = 100, 1000, 10000, 100000$) for sets X of n elements ($n = 3, 4, 5, 6, 7, 8$). In addition, the algorithms presented in subsection 5.2 also depends on the number of elements in the sequence of total orders; therefore, to test the influence of this parameter, the number of elements in the sequence is varied from 100 to 10000.

Let us first study the computation time of the different methods. The results are depicted in Figures **??** and **??**. In these figures, the cardinality of X set is given on the $x-$axis, and the computation time in seconds (in logarithmic scale) is represented on the y-axis.

Figure **??** shows the breakdown of the computation time for Markov-based methods when 10000 linear extensions are generated. It can be seen that there is a severe rise in computation time when n grows. In addition, the Markov-based algorithms dramatically increase their running time if many iterations are considered.

The Markov-based methods with 100 and 10 000 iterations are then compared to the other procedures and the results are shown in Figure **??**. The overall trend shows that minimal elements-based methods and Markov-based method with 100 elements in the sequence are the quickest algorithms.

Let us turn now to study the performance of the different procedures in terms of randomness. For this, we will use the fact that the center of gravity of $\mathscr{FM}(X)$ is a symmetric and autodual fuzzy measure [26]. Thus, the arithmetic mean of the generated measures should be "near" symmetry and "near" duality. To measure to what extent the arithmetic mean is symmetric and autodual the following coefficients are introduced.

Definition 1. *Given $\mu \in \mathscr{FM}(X)$ with X a set of n elements, the asymmetry of μ is computed by*

$$\alpha := \sum_{k=1}^{n-1} \sum_{|A|=k} \frac{(\mu(A) - m_k)^2}{m_k}$$

being

$$m_k = \frac{\sum_{|B_j|=k} \mu(B_j)}{\binom{n}{k}}.$$

This coefficient compares the values of μ with the values of the symmetric measure whose value for cardinality i is the arithmetic mean of the values of μ for the subsets of cardinality i. If μ is symmetric, then $\alpha = 0$.

Definition 2. *Given $\mu \in \mathscr{F}\mathscr{M}(X)$ with X a set of n elements, $A \in \mathscr{P}(X)$, and $A^c \in \mathscr{P}(X)$ its complementary set, the non-duality coefficient is defined by*

$$\beta := \sum_{|A| \leq \frac{n}{2}} (\mu(A) + \mu(A^c) - 1)^2$$

If the fuzzy measure is dual, then $\beta = 0$. Therefore, the coefficients α and β can be used for studying the behavior of the different methods.

Tables 4 and 5 show the Asymmetry and Aduality coefficients obtained when 100000 fuzzy measures are randomly generated with the different methods.

EN MEDIA. DE CUANTO ES LA MUESTRA?

(a)			Asymmetry Coefficient			
Set	M_{100}	M_{1000}	M_{10000}	M^k_{100}	M^k_{1000}	M^k_{10000}
n=3	4.70019E-05	9.731E-08	9.87547E-07	0.000210	6.54771E-07	2.02689E-06
n=4	0.196029	4.81048E-05	6.09445E-07	0.236663	0.000245	2.66345E-06
n=5	1.044679	0.403790	1.03026E-05	1.068692	0.506817	0.000150
n=6	2.737280	2.289562	0.756307	2.744112	2.343422	0.985343
n=7	6.247862	5.996315	4.852164	6.2501405	6.011394	4.988372
n=8	13.479764	13.348894	12.697784	13.479983	13.350559	12.729329
(b)			Aduality Coefficient			
n=3	3.28162E-06	5.76293E-07	1.41061E-06	1.13779E-06	6.74445E-07	3.28079E-06
n=4	1.46596E-06	2.78668E-06	9.53018E-07	1.01014E-06	7.17471E-07	6.30367E-07
n=5	2.04764E-06	2.02489E-06	2.28203E-06	1.65566E-06	1.73626E-06	3.8517E-06
n=6	7.74823E-06	8.31169E-05	0.000232451	2.94934E-07	9.32688E-07	2.61975E-06
n=7	3.7249E-06	6.32291E-06	6.82139E-05	1.23113E-07	4.82296E-07	1.84121E-06
n=8	1.17444E-07	4.59858E-07	5.92201E-06	5.32532E-07	1.31448E-06	4.57988E-07

Table 4: Markov-based measures: (a) Asymmetry Coefficient and (b) Aduality Coefficient

From Tables 4 and 5 it is possible to conclude the following:

- Markov-based methods may not be symmetrical when the number of iterations is low. In addition, the higher n is, the higher must be the number of initial extensions discarded to obtain a symmetrical fuzzy measure. Indeed, for $n \geq 6$, it seems that more than 10 000 iterations are needed.

- The arithemtic mean of the measures generated by the different procedures is almost autodual.

Finally, we have compared the arithmetic mean of the different heuristics with the arithmetic mean of the exact method explained in Section 3. These results are given in Table 6.

(a)			Asymmetry coefficient		
Set	De Loof	Minimals	Minimals$_p$	Lerche	Lerche$_p$
n=3	1.61288E-06	9.29231E-07	3.4345E-07	7.2928E-07	9.9637E-07
n=4	1.54499E-06	4.06191E-06	1.4219E-06	1.88369E-06	9.996E-07
n=5	2.86992E-06	5.70944E-06	2.7828E-06	7.3685E-06	3.5237E-06
n=6	NO	7.3777E-06	6.7548E-06	1.0477E-05	8.82691E-06
n=7	NO	2.15416E-05	7.59801E-06	2.26005E-05	3.20185E-05
n=8	NO	3.78938E-05	2.02846E-05	NO	NO
(b)			Aduality coefficient		
n=3	6.47939E-07	2.55512E-06	4.62077E-06	1.01576E-06	5.57136E-07
n=4	2.70921E-06	5.03652E-05	0,00014979	2.84594E-06	9.79596E-07
n=5	2.39467E-06	0.000193112	0,00041400	7.30625E-06	3.80336E-06
n=6	NO	0.000598669	0.001371804	9.50409E-06	7.12665E-06
n=7	NO	0.001530	0.002262	1.59316E-05	1,46944E-05
n=8	NO	0.003482	0.003379	NO	NO

Table 5: Other methods: (a) Asymmetry Coefficient. (b) Aduality Coefficient

In Table 6 we have discarded Markov methods because they are not symmetric when the number of iterations is reduced. The results in this Table show that although *Minimals* and *Lerche* obtain symmetric and autodual measures, the measure obtained diverges from the measure obtained in the exact procedure. Thus, $Minimals_p$ and $Lerche_p$ seem to be better options in this aspect.

From these results it is possible to conclude that:

- Although Lerche methods are almost dual and symmetrical, their computation time makes the methods unfeasible for the order polytope $\mathcal{FM}(X)$.
- The computation time of Markov based methods with a low number of initial runs (100) is not high, but these methods are not symmetrical, so they do not seem to be a good choice either.
- Minimal elements methods produce symmetric and dual fuzzy measures with the lowest computation time.
- $Minimals_p$ and $Lerche_p$ lead to the same results obtained by the exact procedure when it is possible to apply this last one.

As a conclusion, $Minimals_p$ seems to be the best choice for our purpose among all procedures studied in the paper.

7. Conclusions and open problems

In this paper we have addressed the problem of random generation of fuzzy measures. This problem is interesting when testing different methods of identification of fuzzy measures.

First, we have studied a uniform algorithm. However, this algorithm requires a lot of computation time and is unfeasible for big cardinalities of the referential set. Then, we have proposed a heuristic

	DeLoof	Minimals	Minimals$_p$	Lerche	Lerchep
			$n = 3$		
k=1	0.29763	0.31003	0.29822	0.30944	0.29305
k=2	0.70219	0.69057	0.70282	0.69038	0.70698
			Standard Deviation		
minimum	0.00083	0.00055	0.00010	0.00009	0.00035
maximum	0.00035	0.00041	0.00040	0.00060	0.00061
			$n = 4$		
k=1	0.18097	0.20241	0.18706	0.20220	0.18269
k=2	0.50026	0.49842	0.49547	0.50007	0.50002
k=3	0.81938	0.79978	0.81979	0.79785	0.81746
			Standard Deviation		
minimum	0.00034	0.00037	0.00027	0.00022	0.00017
maximum	0.00046	0.00070	0.00039	0.00054	0.00038
			$n = 5$		
k=1	0.10870	0.13382	0.11789	0.13275	0.11606
k=2	0.34641	0.36066	0.34697	0.36332	0.35565
k=3	0.65343	0.63729	0.64779	0.63643	0.64437
k=4	0.89129	0.87163	0.89243	0.86697	0.88400
			Standard Deviation		
minimum	0.00028	0.00024	0.00017	0.00020	0.00017
maximum	0.00035	0.00059	0.00041	0.00069	0.00046

Table 6: Value (in average) of the fuzzy measures for the subsets of X with cardinality ranging from 1 to $n-1$

based on the concept of minimal elements. This heuristic seems to work in a similar way as the uniform procedure and its computation time reduces a great deal.

Finally, we have also studied the performance of other methods appearing in the literature to deal with this problem. From the simulations carried on, it seems that their behavior is worse than the behavior of the minimal elements procedure for $\mathcal{FM}(X)$.

There are some interesting problems that we pretend to treat in the future:

- We have seen that Markov-based methods have a theoretical background showing that they lead to a random linear extension in limit. From the simulations, it seems that when the cardinality of the referential set increases, the number of iterations needed to obtain something near uniformity grows dramatically. We intend to study in more depth the number of iterations needed in relation with the cardinality of the referential set.

- We have treated the problem of generating fuzzy measures as a special case of generating points in an order polytope. However, we can also treat the problem just considering the concrete structure of $\mathcal{FM}(X)$. This may lead to new procedures with good performance.

8. Acknowledgements

This research has been supported in part by grant numbers MTM2009-10072 and BSCH-UCM910707, and by MEC and FEDER grant TIN2010-14971.

References

[1] G. Beliakov, R. Mesiar, and L. Valášková. Fitting generated aggregation operators to empirical data. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(2):219–236, 2004.

[2] G. Birkhoff. *Lattice Theory, 3rd ed.* Vol. 25 of AMS Colloquium Publications. American Mathematical Society, 1967.

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

[4] R. Bubley and M. Dyer. Faster random generation of linear extensions. *Discrete Mathematics*, 20:81–88, 1999.

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

[6] E. F. Combarro and P. Miranda. Identification of fuzzy measures from sample data with genetic algorithms. *Computers and Operations Research*, 33(10):3046–3066, 2006.

[7] E. F. Combarro and P. Miranda. Adjacency on the order polytope with applications to the theory of fuzzy measures. *Fuzzy Sets and Systems*, 180:384–398, 2010.

[8] E. F. Combarro and P. Miranda. Characterizing isometries on the order polytope with an application to the theory of fuzzy measures. *Information Sciences*, 180:384–398, 2010.

[9] R. Dedekind. Über Zerlegungen von Zahlen durch ihre grössten gemeinsamen Teiler. *Festschrift Hoch Braunschweig Ges. Werke*, II:103–148, 1897. In German.

[10] D. Denneberg. *Non-additive measures and integral*. Kluwer Academic, Dordrecht (the Netherlands), 1994.

[11] L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.

[12] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap. *Aggregation functions*. Cambridge University Press, 2009.

[13] M. Grabisch, T. Murofushi, and M. Sugeno, editors. *Fuzzy Measures and Integrals- Theory and Applications*. Number 40 in Studies in Fuzziness and Soft Computing. Physica–Verlag, Heidelberg (Germany), 2000.

[14] M. Grabisch and J.-M. Nicolas. Classification by fuzzy integral-performance and tests. *Fuzzy Sets and Systems, Special Issue on Pattern Recognition*, (65):255–271, 1994.

[15] M. Huber. Fast perfect sampling from linear extensions. 2006.

[16] A. N. Karkishchenko. Invariant Fuzzy Measures on a Finite Algebra. In *Proc. of the North American Fuzzy Information Processing (NAPIF'96)*, pages 588–592, 1996.

[17] A. Karzanov and L. Khachiyan. On the conductance of order Markov chains. *Order* 8(1):7-15, 1995.

[18] G. Koshevoy. Distributive lattices and product of capacities. *Journal of Mathematical Analysis and Applications*, 219:427–441, 1998.

[19] K.-M. Lee and H. Leekwang. Identification of λ-measures by genetic algorithms. *Fuzzy Sets and Systems*, (75):301–309, 1995.

[20] D. Lerche, P. Sörensen. Evaluation of the ranking probabilities for partial orders based on random linear extensions. *Chemosphere*, 53:981–992, 2003.

[21] D. Lerche, P. Sörensen, R. Bruggemann. Improved estimation of the ranking probabilities in partial orders using random linear extensions by approximation of the mutual ranking probability. *J. Chem. Inf. Comput. Sci.*, 43:1471–1480, 2003.

[22] J. Leydold and W. Hörmann. A Sweep-Plane Algorithm for Generating Random Tuples in Simple Polytopes. *J. Math. Comp.*, (67):1617–1635, 1998.

[23] K. De Loof, H. De Meyer and B. De Baets. Exploiting the lattice of ideals representation of a poset. *Fundam. Inf.*, 71(2,3):309–321, 2006.

[24] L. Nourine M. Habib, R. Medina and G. Steiner. Efficient algorithms on distributive lattices. *Discrete Applied Mathematics*, 110:169–187, 2001.

[25] J. Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

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

[27] P. Miranda, M. Grabisch and P. Gil. Identification of non-additive measures from sample data. In R. Kruse G. Della Riccia, D. Dubois and H.-J. Lenz, editors, *Planning based on Decision Theory*, volume 472 of *CISM Courses and Lectures*, pages 43–63. Springer Verlag, 2003.

[28] R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*. Wiley, 2007.

[29] E. Schmerling. An Algorithm for Generating Sequences of Random Tuples on Special Simple Polytopes. In *Proc. of the Int. Conf. on Computational and Math. Methods in Sci. and Engineering (CMMSE2009)*, pages 989–994, Gijón (Spain), July 2009.

[30] R. Stanley. Two poset polytopes. *Discrete Comput. Geom.*, 1(1):9–23, 1986.

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

[32] E. Takahagi. A fuzzy measure identification method by diamond pairwise comparisons: AHP scales and Grabisch's graphical interpretation. In *Knowledge-Based Intel. Inf. and Engeneering Systems*, volume 4694 of *Lecture Notes in Computer Science*, pages 316–324. Springer, 2009.

[33] L. Valiant. The complexity of enumerating and reliability problems. *SIAM Journal of Computing*, (8):410–421, 1979.

[34] S. T. Wierzchoń. An algorithm for identification of fuzzy measures. *Fuzzy Sets and Systems*, (9):69–78, 1983.

[35] D. B. Wilson Mixing times of lozenge tiling and card shuffling Markov chains. *Ann. Appl. Probab.*, 14(1):274-325, 2004.