Information Sciences 659 (2024) 120080

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Random generation of linearly constrained fuzzy measures and domain coverage performance evaluation

Jian-Zhang Wu [a,*], Gleb Beliakov [a], Simon James [a], Marek Gagolewski [a,b,c]

[a] School of Information Technology, Deakin University, Burwood, 3125, Australia
[b] Warsaw University of Technology, Faculty of Mathematics and Information Science, ul. Koszykowa 75, 00-662 Warsaw, Poland
[c] Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland

ARTICLE INFO

Keywords:
Fuzzy measure
Random generation
Linear programming
Domain coverage
Convex combination

ABSTRACT

The random generation of fuzzy measures under complex linear constraints holds significance in various fields, including optimization solutions, machine learning, decision making, and property investigation. However, most existing random generation methods primarily focus on addressing the monotonicity and normalization conditions inherent in the construction of fuzzy measures, rather than the linear constraints that are crucial for representing special families of fuzzy measures and additional preference information. In this paper, we present two categories of methods to address the generation of linearly constrained fuzzy measures using linear programming models. These methods enable a comprehensive exploration and coverage of the entire feasible convex domain. The first category involves randomly selecting a subset and assigning measure values within the allowable range under given linear constraints. The second category utilizes convex combinations of constrained extreme fuzzy measures and vertex fuzzy measures. Then we employ some indices of fuzzy measures, objective functions, and distances to domain boundaries to evaluate the coverage performance of these methods across the entire feasible domain. We further provide enhancement techniques to improve the coverage ratios. Finally, we discuss and demonstrate potential applications of these generation methods in practical scenarios.

1. Introduction

Fuzzy measures and fuzzy integrals have been established as useful tools for representing interaction among multiple items or criteria and for aggregating the interdependent multiple sources information. [8,27]. Although nonadditivity and nonmodularity commonly exist among different types of fuzzy measures, most of these flexible constraints can still be represented into linear forms. Firstly, the normalization conditions and monotonicity requirements with respect to set inclusion for defining normal fuzzy measures can be rewritten as a bunch of linear constraints. Secondly, some special types of fuzzy measures can be identified through a set of linear, 0-1 or even integer constraints, like the k-additive fuzzy measures [13], p-symmetric fuzzy measures [21], k-maxitive and minitive fuzzy measures [23], k-interactive fuzzy measure and so on. Thirdly, fuzzy measures have many equivalent representations – Möbius representation [7], the Shapley and interaction indices [17], the nonadditivity and nonmodularity indices [29] – all of which can be obtained through invertible linear transformations. Furthermore, the comparison between fuzzy integrals including at least Choquet integrals [8] and Sugeno integrals [27] of given alternatives can be constructed as linear conditions.

* Corresponding author.
E-mail address: jianzhang.wu@deakin.edu.au (J.-Z. Wu).

https://doi.org/10.1016/j.ins.2023.120080
Received 2 June 2023; Received in revised form 7 November 2023; Accepted 28 December 2023

Available online 4 January 2024

All of the constraints mentioned above will ultimately constitute a convex domain of feasible fuzzy measures, providing us with an arena to explore various optimization problems, find pattern and knowledge through machine learning, evaluate and compare alternatives under multiple criteria decision environment and investigate the properties and connections among measures, integral as well as their indices.

Compared to traditional optimization methods, random generation has some advantages in exploring entire domain space, overcoming local optima, computing efficiency and adaptability. We can broadly classify these methods of random fuzzy measure generation into two general approaches: random-node based methods and extreme measures based methods. The former methods randomly choose a subset and then randomly identify a value in its feasible range, which will be between the maximum value assigned to its proper subsets and the minimum of its supersets [9,10,18]. The latter methods involve the random generation of fuzzy measures through the utilization of randomized convex combinations of extreme fuzzy measures. This approach often preserves crucial properties across subsequent generations, making it highly valuable in generic algorithms and other machine learning techniques [4,9].

However, most existing random generation methods only consider the first type of conditions mentioned above, namely the boundary and monotonicity conditions on fuzzy measures. As a result, random-node based methods and extreme measures based methods cannot be directly used for more complex linear constraints. This is because, on the one hand, the feasible range of a subset's measure is no longer entirely determined by the minimum of its supersets' measures and the maximum of its proper subsets' measures. On the other hand, the extreme fuzzy measures may not always belong to the linearly constrained feasible domain, let alone their convex combinations.

In this paper, a key contribution is the extension of these two primary methods for random generation of fuzzy measures using linear programming techniques, which can effectively incorporate various linear constraints mentioned above. For the random-node based methods, linear programming models are adopted to further refine the close intervals over which fuzzy measure value of selected subsets can be taken. For the extreme measures methods, linear programming models are applied to determine the constrained extreme fuzzy measures satisfying all the linear constraints while pursuing the maximum or minimum measure values. Instead of attempting to find all possible extreme fuzzy measures, known as the Dedekind numbers [14], which can quickly become a time-consuming and unattainable task, even for just a few variables, we randomly select a relatively small number of extreme fuzzy measures in each loop and ensure to efficiently generate fuzzy measures without sacrificing much accuracy.

Furthermore, by utilizing the convexity property of linear programming models and convex combination operation, we propose a third approach for generating fuzzy measures, called the vertices measures based method. The vertex measures refer to the fuzzy measures that lie on the intersection of two or more boundary facets of the feasible convex domain. These fuzzy measures can be obtained by randomly selecting a few linear constraints and minimizing their deviation variables through a multiple goal linear programming model. It should be mentioned that these vertex measures are not necessarily the basic solutions in the convex domain, and therefore the computation cost of checking the linear independence among the selected constraints can be omitted. Furthermore, these vertex fuzzy measures have the advantage of generating approximate solutions around the optimal fuzzy measures for linear objective functions.

Another key contribution of this paper is to establish performance evaluation methods for the generated fuzzy measures to cover the entire feasible domain. We utilize three types of techniques to describe the coverage. The first technique uses different types of indices of fuzzy measures, such as orness index [12], entropy value [20], and even integral value of a typical alternative [8]. The second technique considers the objective function of the optimization model used in identifying fuzzy measures, such as least-squares methods [15,16,24], least absolute deviation methods [1], maximum split methods [22,24], maximum entropy methods [19,20], and compromise principle [32]. The third technique adopts the distances of fuzzy measures to the boundaries of the feasible domain, which can be obtained from multiple goals linear programming models. All the ranges of these distances, indices, and objective functions under the linear constraints can provide direct and distinct ways to evaluate the coverage of generated fuzzy measures.

The paper is organized as follows. Following the introduction, Section 2 provides the necessary background knowledge on fuzzy measures, including linear transformations, the Choquet integral, and the linear constraints relevant to the mentioned indices. Section 3 focuses on convex combinations and presents three types of linear programming-based random generation methods for fuzzy measures. The coverage performance evaluation of the generated fuzzy measures is discussed in Section 4. Several examples are used to demonstrate the random generation algorithms in Section 5. Section 6 introduces approaches for enhancing the coverage ratios. Furthermore, we list some potential applications of the proposed methods. Finally, the paper concludes in Section 8.

2. Preliminaries

Let $N = \{1, 2, \dots, n\}$, $n \geqslant 2$, be the set of elements, $\mathcal{P}(N)$ be the power set of N, and $|S|$ be the cardinality of subset $S \subseteq N$.

Definition 1. [8] A fuzzy measure on N is a set function $\mu : \mathcal{P}(N) \to [0, 1]$ such that (i) $\mu(\emptyset) = 0$, $\mu(N) = 1$; (ii) $\forall A, B \subseteq N$, $A \subseteq B$ implies $\mu(A) \leqslant \mu(B)$.

The value of $\mu(A)$ reflects the importance of subset A in the decision context.

The interaction phenomenon between multiple elements is conventionally described by means of the simultaneous interaction indices [17], including the Möbius representation [7] and Shapley importance and interaction index [13].

Definition 2. [7] Let μ be a fuzzy measure on N, the Möbius representation for each set is given by

$$m_\mu(A) = \sum_{B\subseteq A} (-1)^{|A\setminus B|}\mu(B), A \subseteq N.$$

Möbius representation can also be referred to as the internal simultaneous interaction index for each coalition.

Definition 3. [17] Let μ be a fuzzy measure on N, its Shapley importance and interaction index is defined as

$$I_\mu(A) = \sum_{B\subseteq N\setminus A} \frac{1}{(|N|-|A|+1)\binom{|N|-|A|}{|B|}} \sum_{C\subseteq A} (-1)^{|A\setminus C|}\mu(C\cup B), A\subseteq N.$$

Generally speaking, $I_\mu(\{i\})$ is regarded as the overall importance of $i\in N$, and $I_\mu(A)$, $|A|\geqslant 2$ provide the simultaneous interaction indices for each subset A, involving both the internal and external subsets.

The above two indices can rightly be considered as alternative representations of μ, defined as they are by invertible one-to-one linear mappings (see [4,14]).

Certain restrictions and construction methods lead to different types of fuzzy measures, sometimes reflecting decision-maker preferences on the aggregation behavior. We make particular mention of the k-additive [13], p-symmetric [25], k-tolerant and -intolerant [21], and k-maxitive and -minitive fuzzy measures [6]. Each offers a reduction in the number of variables required to completely define the fuzzy measure values. For example, k-additive fuzzy measures are defined as follows.

Definition 4. [13] Let $k\in\{1,...,n\}$, a fuzzy measure μ on N is said to be k-additive if its Möbius representation satisfies $m_\mu(A)=0$ for all $A\subseteq N$ such that $|A|>k$ and there exists at least one subset A of k elements such that $m_\mu(A)\neq 0$.

It is hence defined by a combination of linear constraints in Möbius representation or equivalent linear constraints in other representations.

The Choquet integral is a widely studied fuzzy integral that can be used to aggregate a vector of inputs according to the importance and interaction information contained in the defining fuzzy measure.

Definition 5. [28] Let $\mathbf{x}$ be a real-valued vector function on N, $\mathbf{x}:=(x_1,\dots,x_n)\in[0,1]^n$. The Choquet integral of $\mathbf{x}$ with respect to a fuzzy measure μ on N is defined as

$$C_\mu(\mathbf{x}) = \sum_{i=1}^{n} [x_{(i)} - x_{(i-1)}]\mu(N_{(i)}) = \sum_{i=1}^{n} [\mu(N_{(i)}) - \mu(N_{(i+1)})]x_{(i)} \tag{1}$$

where the parentheses used for indices represent a permutation on N such that $x_{(1)}\leqslant\cdots\leqslant x_{(n)}$, $x_{(0)}=0$, $N_{(i)}=\{(i),\dots,(n)\}$, and $N_{(n+1)}=\emptyset$.

If $\mathbf{x}$ is fixed, the Choquet integral amounts to a linear combination of the fuzzy measure values.

The orness index [12] associated with an aggregation function describes the tendency toward max-like behavior. In the case of the Choquet integral, orness can be calculated directly from the fuzzy measure values.

Definition 6. [21] The orness index of fuzzy measure μ is given by

$$orness(\mu) = \sum_{A\subset N} \frac{(n-|A|)!|A|!}{n!(n-1)}\mu(A). \tag{2}$$

Besides the orness index, another famous index used to describe the fuzzy measure behavior is entropy.

Definition 7. [20] The entropy of a fuzzy measure μ on N is given by

$$E(\mu) = \sum_{i=1}^{n} \sum_{A\subseteq N\setminus\{i\}} \frac{(|N|-|A|-1)!|A|!}{|N|!} h\left(\mu(A\cup\{i\})-\mu(A)\right), \tag{3}$$

where $h(x)=-x\ \ln(x)$ if $x>0$ and 0 if $x=0$.

Maximum entropy is obtained for any n if the fuzzy measure is symmetric and additive, at which point it collapses to the arithmetic mean [5,32].

For a fixed N we use $\mathcal{M}$ to denote the set of valid fuzzy measures, which forms a convex polytope on $[0,1]^{2^n-2}$. Most existing random fuzzy measure generation methods only involve restrictions that model the monotonicity conditions [2,3], i.e.,

$$\mu(B\cup\{i\})-\mu(B)\geqslant 0, \forall i\in N, \forall B\subseteq N\setminus i, \tag{4}$$

however in practical applications some additional preference information may further be considered. We use the following conventional comparison and interval forms to represent additional constraints on criteria and alternatives [4,15]:

- the overall importance of element i is at least as great as that of j:

$$I_\mu(\{i\}) - I_\mu(\{j\}) \geqslant 0,$$

- the comprehensive interaction of coalition $\{i,j\}$ is positive or negative:

$$\delta \leqslant I_\mu(\{i,j\}) \leqslant 1 \text{ or } -1 \leqslant I_\mu(\{i,j\}) \leqslant -\delta,$$

- the comprehensive interaction of subset A is greater than that of B:

$$I_\mu(A) - I_\mu(B) \geqslant \delta,$$

- alternative **a** is as good as **b**:

$$C(\mathbf{a}) - C(\mathbf{b}) \leqslant \delta \text{ and } C(\mathbf{a}) - C(\mathbf{b}) \geqslant -\delta,$$

- the fuzzy measure should be 2-additive:

$$m_\mu(A) = 0, |A| > 2,$$

where δ is a small positive threshold, e.g., 0.05.

All of the above constraints are linear and constitute a convex feasible range if they are compatible. Otherwise, some multiple goal linear programming based methods and strategies can aid to recognize the existing contradictions and adjust them into a consistent case [30].

3. Linear programming extended random generation methods

In this section, we aim to extend the existing approaches of random generation of fuzzy measures, such as the random node-based and extreme measures-based methods, to deal with more complex linear constraints. Specifically, we propose to use linear programming techniques and convex combinations of fuzzy measures to efficiently generate ones that satisfy a larger set of linear constraints beyond just the boundary and monotonicity conditions.

3.1. Update random node-based methods by linear programming models

A straightforward approach to generating a random fuzzy measure on N is to first randomly select a subset (node) from N and then identify a random value within its allowable range [9,18]. This approach under normalization and monotonicity conditions is shown in Algorithm 1.

Algorithm 1: Random node-based generation methods under normalization and monotonicity conditions.

```
Input: k̄ and N.
for k in 1 : k̄ do
    μ_k(N) = 1 and μ_k(∅) = 0.
    Let P⁺ denote all the nonempty proper subsets of N,
    Let P⁻ = {∅, N},
    for i in 1 : (2^n − 2) do
        Randomly choose a subset A ∈ P⁺,
        Calculate L_A = max_{B⊂A, B∈P⁻} μ_k(B),
        Calculate U_A = min_{C⊃A, C∈P⁻} μ_k(C),
        Set μ_k(A) to a random value in [L_A, U_A],
        Set P⁺ = P⁺ \ A, P⁻ = P⁻ ∪ A.
    end
    Store μ_k, as the k-th row of matrix U.
end
Return matrix U_{k̄×2^N} of k̄ fuzzy measures.
```

The normalization or boundary conditions can be relatively trivial by setting the measure values of the empty set and the universal set as 0 and 1, respectively. The main challenge is to ensure the monotonicity condition with respect to any set inclusion. This is achieved by searching for the allowance interval $[L_A, U_A]$ under the predefined measure values of all sets in $\mathcal{P}^-$.

For convenience, we denote the boundary and monotonicity conditions as $\mathcal{B}$ and additional linear constraints as $\mathcal{D}$. In this subsection, we extend Algorithm 1 into a more general form to deal with any linear constraints within $\mathcal{D}$, see Algorithm 2. Given that the conditions of $\mathcal{B}$ and $\mathcal{D}$ are linear, and if they are compatible, the joint feasible domain must be a convex region. Methods for inconsistency checking and improvement can be found in [30,31]. This convexity ensures that the allowable range of $\mu(A)$ is always

Algorithm 2: Linear programming updated random node generation method.

```
Input: k̄ and N.
for k in 1 : k̄ do
    μ_k(N) = 1 and μ_k(∅) = 0.
    Let 𝒫 denote all the nonempty proper subsets of N,
    for i in 1 : (2^n − 2) do
        Randomly choose a subset A ∈ 𝒫,
        Calculate L_A = min μ_k(A) s.t. ℬ and 𝒟,
        Calculate U_A = max μ_k(A) s.t. ℬ and 𝒟,
        Set μ_k(A) to a random value r in [L_A, U_A],
        Set 𝒫 = 𝒫 \ A, and add constraint, μ_k(A) = r, into 𝒟.
    end
    Store μ_k, as the k-th row of matrix U.
end
Return matrix U_{k̄×2^N} of k̄ fuzzy measures.
```

a closed interval. This means that there is still some degree of randomness' freedom in generating measure values by using linear programming models.

The extension of Algorithm 1 to Algorithm 2 with $\mathcal{D} \neq \emptyset$, allows for the incorporation of additional linear constraints, making it more flexible and applicable to a wider range of scenarios. Furthermore, the maintenance of linearity and convexity ensures that the Algorithm 2 can be directly used with any linear equivalent representation of fuzzy measures, such as the Möbius representation, the Shapley interaction index, nonadditivity index, and nonmodularity index, which is a significant advantage in terms of applicability.

3.2. Convex combination and constrained extreme fuzzy measures

The feasible range determined by a set of linear constraints is a convex set, and hence convex combinations of feasible solutions will also be feasible. The convex combination of fuzzy measures $\mu_1, \ldots, \mu_t$, $t > 1$, is defined as [9]:

$$\lambda_1\mu_1 + \cdots + \lambda_t\mu_t, \quad \sum_{i=1}^{t} \lambda_i = 1, \quad \lambda_i \geqslant 0. \tag{5}$$

This formula allows any feasible fuzzy measure to serve as the parent measure for generating new fuzzy measures between them. However, one of the ultimate goals of a random generation method is to cover the entire feasible range as much as possible. In this context, the use of extreme fuzzy measures, which usually correspond to the boundary points of the polytope of fuzzy measures, can provide a more significant coverage ratio of the feasible domain.

Only under the boundary and monotonicity conditions, the extreme values of $\mu(A)$ can reach 0 and 1 in two opposite directions and one basic fact is that any fuzzy measure is a convex combination of 0-1 valued extreme fuzzy measures [14]. The following theorem provides a possible convex combination of 0-1 valued measures for a fuzzy measure.

Theorem 1. *A fuzzy measure μ on N is a convex combination of $|\{\mu(A), A \subseteq N, \mu(A) > 0\}|$ extreme fuzzy measures on N.*

Proof. A straightforward way to illustrate this is via a relation $\prec$ on $\mathcal{P}(N)$ corresponding with increasing fuzzy measure values, i.e., $A \prec B$ if $\mu(A) < \mu(B)$, and the following construction. Let $x_{(i)}$, $i = 1, \ldots k$ denote the k non-zero unique fuzzy measure values in the increasing order. For a fuzzy measure μ, we use $\mu_{(i)}$ to denote the extreme fuzzy measure corresponding with $x_{(i)}$, such that there is some $\mu(B) = x_{(i)}$ and $\mu_{(i)}(A) = 0$ for all $A \prec B$ and 1 otherwise. Then $\mu = \sum_{i=1}^{k} \lambda_i \mu_{(i)}$, with $x_{(i)} = \sum_{j=1}^{i} \lambda_j$. □

The number of extreme fuzzy measures that take only 0 or 1 as their values are given by $M(n) - 2$, where $M(n)$ is a Dedekind number that grows exponentially with n [2,10,11,14]. As n becomes large, it becomes computationally impractical to generate all such extreme points. Fortunately, Theorem 1 suggests that any fuzzy measure can be expressed as a convex combination of different extreme fuzzy measures, making it unnecessary to loop through all extreme fuzzy measures when using random generation methods.

Let us consider the case where $\mu(A) = 1$. Based on the monotonicity condition of set inclusion, we can obtain the following necessary condition:

$$\mu(B) = 1, \forall B \cap A = A. \tag{6}$$

In another extreme case, we can assume that any part of the subset A can result in the belonging coalition having a fuzzy measure value of 1. In other words, we can have:

$$\mu(B) = 1, \forall B \cap A \neq \emptyset. \tag{7}$$

These two cases correspond to the necessity and possibility measure focused on A [26], which are considered as the fundamental types of extreme measures in random generation methods.

More generally, if the linear constraints include more than the boundary and monotonicity conditions, the extreme values of $\mu(A)$ may not necessarily be 0 or 1. However, drawing inspiration from Eqs. (6) and (7), we can formulate the following two linear models to generate constrained extreme fuzzy measures:

$$\max \sum_{\forall B\cap A=A} \mu(B) - \sum_{\forall B\cap A\neq A} \mu(B) \quad \text{s.t. } \mathcal{B} \text{ and } \mathcal{D}. \tag{8}$$

$$\max \sum_{\forall B\cap A\neq\emptyset} \mu(B) - \sum_{\forall B\cap A=\emptyset} \mu(B) \quad \text{s.t. } \mathcal{B} \text{ and } \mathcal{D}. \tag{9}$$

For convenience, we can refer to the optimal solutions of the above models (8) and (9) as the constrained necessity measure and the constrained possibility measure, respectively. It is obvious that if $\mathcal{D}=\emptyset$, the constrained necessity and possibility measures are just the 0-1 valued necessity and possibility measures. Based convex combinations of these constrained extreme fuzzy measure, we can have the following Algorithm 3.[1]

Algorithm 3: Random generation method of constrained extreme fuzzy measures and convex combination.

```
Input: k̄ and N.
for any nonempty subset A in N do
    Get the constrained necessity and possibility measures by Models (8) and (9),
    Store these two measures as two rows in the matrix V.
end
for k in 1 : k̄ do
    Randomly select t, 2 ⩽ t ⩽ |V|, extreme fuzzy measures in V, denoted υ_1, ..., υ_t.
    Randomly generate t positive numbers λ_1, ..., λ_t with their sum equal to 1.[1]
    Set μ_k = λ_1 υ_1 + ⋯ + λ_t υ_t, and store μ_k as the k-th row of U.
end
Return matrix V_{k̄×2^N} and U_{k̄×2^N} of 2 × k̄ fuzzy measures.
```

Equations (8) and (9) suggest that the constrained extreme measures are designed to achieve the upper and lower limits of measure values for subsets. From the perspective of linear constraints other than measure values, there is another category of extreme fuzzy measures that exist at the intersection of certain facets of the feasible domain, referred to as vertex fuzzy measures. This category of extreme fuzzy measures can lead to the development of another set of random generation methods.

3.3. Vertex fuzzy measures based random generation

Taking the fuzzy measure values of 2^n subsets as variables, the normalization, monotonicity, and additional linear constraints can be universally represented in matrix form as:

$$\mathbf{A}(\mu(\emptyset), \mu(\{1\}), \ldots, \mu(N)))^{\mathrm{T}} = (\geqslant, \leqslant)\mathbf{b}, \tag{10}$$

where $\mathbf{A} = [a_{ij}]_{q\times 2^n}$ is a coefficient matrix, q is the total number of constraints, and $\mathbf{b}_{q\times 1} = (b_1, \ldots, b_q)^{\mathrm{T}}$ is the right-hand side vector.

We further categorize all constraints into three types based on their direction of inequality as $Q_=, Q_\geqslant$, and $Q_\leqslant$. By introducing deviation variables, we can formulate a multiple goals linear program to obtain the vertex fuzzy measures of Eq. (10):

$$\begin{aligned}
&\min z = \sum_{i\in Q'} d_i \\
&s.t.\ a_{i0}\mu(\emptyset) + a_{i1}\mu(\{1\}) + \cdots + a_{i2^n}\mu(N) = b_i, i \in Q_=, \\
&a_{i0}\mu(\emptyset) + a_{i1}\mu(\{1\}) + \cdots + a_{i2^n}\mu(N) - d_i = b_i, i \in Q_\geqslant, \\
&a_{i0}\mu(\emptyset) + a_{i1}\mu(\{1\}) + \cdots + a_{i2^n}\mu(N) + d_i = b_i, i \in Q_\leqslant,
\end{aligned} \tag{11}$$

where d_i are the positive or negative deviation variables, $d_i \geqslant 0$, and $Q' \subseteq \{1, \ldots, q\}$ is a randomly selected subset of the constraints index set, $|Q'| \geqslant 2$. If the deviation variable is zero, then the corresponding constraint will be strict. Taking into account that all the original linear constraints are compatible (they can be redundant but without any contradictions), the optimal solution of the above model must reach the boundary facet and will result in vertex fuzzy measures (the optimal objective function value will not be zero if some redundant constraints exist). Then, utilizing convex combinations, we construct Algorithm 4.

In this section, we present three categories of generation methods for fuzzy measures. Our subsequent objective is to evaluate the quality of the generated results. To achieve this, we introduce the concept of coverage of the feasible domain.

[1] Here, we can generate $t-1$ values with a uniform distribution in the range of $[0,1]$. These values can be sorted in ascending order as $\lambda'_1, \cdots, \lambda'_{t-1}$. Subsequently, we define $\lambda_1 = \lambda'_1$, $\lambda_2 = \lambda_1 - \lambda'_1$, $\cdots$, and $\lambda_t = 1 - \lambda'_{t-1}$.

Algorithm 4: Random generation method of vertex fuzzy measures and convex combination.

Input: $\overline{k}$, $\overline{l}$ and N.
for l *in* $1:\overline{l}$ **do**
  Randomly select a subset of the constraints index set, denoted Q',
  Construct the linear program according to (11) and denote the optimal fuzzy measure obtained as μ_l,
  Store μ_l as the l-th row of **V**.
end
for k *in* $1:\overline{k}$ **do**
  Randomly select t extreme fuzzy measure in **V**, denoted $v_1, ..., v_t$
  Randomly generate t positive numbers $\lambda_1, ..., \lambda_t$ with their sum equal to 1.[1]
  Set $\mu_k = \lambda_1 v_1 + \cdots + v_t \lambda_t$, and store μ_k as the k-th row of **U**
end
Return matrix $\mathbf{V}_{\overline{l}\times 2^N}$ of $\overline{l}$ fuzzy measures and $\mathbf{U}_{\overline{k}\times 2^N}$ of $\overline{k}$ fuzzy measures.

4. Evaluating coverage performance of generated fuzzy measures

The coverage of a set of fuzzy measures can be quantified as the proportion of the feasible domain that is covered by the generated fuzzy measures. A high coverage ratio of the generated measures is indicative of a comprehensive representation of the entire feasible domain, rather than a limited and skewed portion of it. Given the high dimensionality and complex shape of the feasible domain with at least 2^n variables and a large number of monotonicity and additional linear constraints, it can be challenging to obtain a comprehensive understanding of the domain range. Therefore, we can employ certain indices to probe the range of the domain and obtain insights into the coverage ratio. Here, we utilize three types of indices to calculate the coverage ratio: the indices of the fuzzy measure, the objective function of the fuzzy measure identification model, and the distance to the boundaries of the feasible domain.

4.1. Measuring coverage by indices of fuzzy measure

The first type includes all indices of the fuzzy measure. These indices can be calculated from linear combinations of the fuzzy measure values, such as the orness index (see Eq. (2)), the Shapley interaction index (see Definition 3), and the Choquet integral output for a given alternative (see Definition 5), as well as some nonlinear representation such as entropy (see Eq. (3)). It should be noted that the fuzzy measure values of any subset except the empty set and the universal set can be taken to measure the coverage.

Only with boundary and monotonicity conditions, the range of orness is $[0,1]$, the range of entropy is $[0,\ln(n)]$, the range of the Choquet integral output for a given $\mathbf{x}$ is $[\min_i x, \max_i x]$, the range of fuzzy measure value of a subset can be $[0,1]$. With the addition of additional linear constraints, the feasible domain remains convex, and the above indices typically have close ranges.

4.2. Measuring coverage by the objective functions of identification models

The identification method obtains the desired fuzzy measures according to an objective function, which may include minimizing or maximizing the indices of fuzzy measures such as orness, entropy, or interaction indices. In some models, the objective function may be more complex by involving the Choquet integral values of given alternatives [4,15,32].

- Minimum deviations between integral values and desired values

$$\min z = \sum_{\mathbf{x}\in L} (C_\mu(\mathbf{x}) - y(\mathbf{x}))^2 \quad \text{s.t. } \mathcal{B} \text{ and } \mathcal{D}. \tag{12}$$

  where L is the set of all alternatives, $y(\mathbf{x})$ is the expected overall evaluation of each $\mathbf{x}\in L$, the objective is to minimize the squared distance between the Choquet integral evaluations and the expected overall evaluations across alternatives in L.
- Maximum split among integral values

$$\begin{aligned} &\max z = \epsilon \\ &\text{s.t. } \mathcal{B} \text{ and } \mathcal{D}, \\ &C_\mu(\mathbf{x}) - C_\mu(\mathbf{x}') \geqslant \epsilon \text{ if } \mathbf{x} \succ \mathbf{x}' \in O(L). \end{aligned} \tag{13}$$

  The above model maximizes the distances among all neighbor alternatives in the given order $O(L)$.
- Maximize the sum of integral values

$$\max z = \sum_{\mathbf{x}\in L} C_\mu(\mathbf{x}) \quad \text{s.t. } \mathcal{B} \text{ and } \mathcal{D}. \tag{14}$$

  The above objective function actually stems from the compromise principle for identifying fuzzy measures [32], which aims to give each alternative an equal chance to reach its highest potential evaluation.

Table 1
Experimental results of coverage ratios on four elements of Algorithm 1 or 2.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
Orness	0.702	0.715	**0.720**	0.720	0.726	0.737	0.010
Entropy	0.604	0.611	**0.618**	0.618	0.626	0.633	0.009
Choquet Integral	0.909	0.920	**0.928**	0.926	0.933	0.935	0.009
Distance (1)	0.881	0.885	**0.895**	0.891	0.896	0.899	0.007
Distance (2)	0.879	0.886	**0.892**	0.891	0.894	0.907	0.009
Distance (3)	0.877	0.888	**0.895**	0.895	0.902	0.909	0.010

4.3. Measuring coverage by distances to boundaries of feasible domain

Each constraint, if not redundant, can form a boundary facet of the feasible domain. To calculate the coverage ratio, we first need to obtain the maximum and minimum distances to each constraint. If a constraint is in $Q_=$, then the distance of any fuzzy measure to it will be zero. For constraints in $Q_\geqslant$ and $Q_\leqslant$, we can use the following formula to calculate the range of distances:

$$\begin{aligned} &\min(\max) z = d_i \\ \text{s.t. } &\mathcal{B} \text{ and } \mathcal{D}, \\ &a_{i0}\mu(\emptyset) + a_{i1}\mu(\{1\}) + \cdots + a_{i2^n}\mu(N) - d_i = b_i, \text{if } i \in Q_\geqslant, \\ &a_{i0}\mu(\emptyset) + a_{i1}\mu(\{1\}) + \cdots + a_{i2^n}\mu(N) + d_i = b_i, \text{if } i \in Q_\leqslant, \end{aligned} \tag{15}$$

For any given fuzzy measure μ, we just use

$$d_i^\mu = |a_{i0}\mu(\emptyset) + a_{i1}\mu(\{1\}) + \cdots + a_{i2^n}\mu(N) - b_i| \tag{16}$$

to calculate its distance to the constraint.

4.4. Indices granularity based coverage ratio

As discussed above, a single index used for evaluating coverage, denoted as z, should lie within a closed interval $[L_z, U_z]$ under a set of linear constraints. We can divide this close interval into a given number of equal sub-intervals, such as $b = 1000$, and calculate the coverage ratio as the ratio of sub-intervals occupied by the generated fuzzy measures out of the total number of sub-intervals, denoted as r_b. It is important to note that the number of generated fuzzy measures should be substantially greater than the number of sub-intervals, as a smaller number of generated measures may result in the coverage ratio never reaching 1.

5. Experimental results

In this section, we use two illustrative examples to show the coverage ratios of the proposed random generation methods.

5.1. Random generation only with boundary and monotonicity conditions

Considering the universal set as $N = \{1, 2, 3, 4\}$, we first investigate the random node-based random generation method, Algorithm 1, which is equivalent to Algorithm 2 when subject only to boundary and monotonicity conditions. To conduct our analysis, we set the number of generated fuzzy measures, denoted as $\overline{k}$, to 5000. Table 1 presents statistical data for coverage ratios obtained from 10 independent algorithm runs. In the table and forthcoming, Min., 1st Qu., Median, Mean, 3rd Qu., Max., and sd. represent minimum, first quartile, median, mean, third quartile, maximum, and standard deviation of the 10 coverage ratios, respectively. The coverage ratios are calculated based on the orness, entropy, Choquet integral of the input vector (0.25, 0.5, 0.75, 1), and the distances to three chosen boundary conditions, corresponding to the second to seventh rows in Table 1 respectively. In this case, the ranges of these indices are as follows: $[0, 1]$ for orness index, $[0, \ln(4)]$ for entropy, $[0.25, 1]$ for Choquet integral, and $[0, 1]$ for the distances. Additionally, it's worth noting that the coverage ratio calculation is based on 1000 breaks, i.e., $b = 1000$.

The random node-based generation method is commonly used as a benchmark for performance comparison. In this study, we use the median of coverage ratio as the main indicator.

Algorithm 3 utilizes convex combinations of extreme fuzzy measures, specifically the 30 0-1 possibility or necessity measures. Table 2 displays the coverage ratios obtained from generating 5000 fuzzy measures for $t = 2, 3$ in 10 independent runs. For $t = 2$, the medians of the coverage ratios for orness and Choquet integral of the input vector (0.25, 0.5, 0.75, 1) exceed those in Table 1. For $t = 3$, the medians of the coverage ratios for orness, entropy, and Choquet integral are higher than those in Table 1. Additionally, when $t = 3$, the coverage ratios for the three distances reach their peak values in Table 2.

Algorithm 4 generates fuzzy measures using convex combinations of vertex fuzzy measures obtained through linear programming. Table 3 displays the results for various parameters, specifically $|Q'| \sim 0.6 \times 2^4 = 10$, $\overline{l} = 100, 150$, and $t = 2, 3$. Each row still corresponds to the coverage ratios of generating 5000 fuzzy measures in 10 independent runs. Among these cases, when $\overline{l} = 150$ and $t = 3$, all the coverage ratios exceed the benchmark values. In the case of $\overline{l} = 150, t = 3$, four coverage ratios outperform their respective

Table 2
Experimental results of coverage ratios on four elements of Algorithm 3.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
$t = 2$	Orness	0.913	0.919	**0.924**	0.924	0.930	0.938	0.008
	Entropy	0.487	0.493	0.494	0.493	0.495	0.497	0.003
	Choquet Integral	0.960	0.963	**0.969**	0.969	0.974	0.980	0.007
	Distance (1)	0.754	0.761	0.769	0.770	0.774	0.793	0.012
	Distance (2)	0.760	0.764	0.772	0.773	0.780	0.788	0.009
	Distance (3)	0.760	0.762	0.774	0.773	0.781	0.788	0.011
$t = 3$	Orness	0.834	0.838	**0.847**	0.845	0.852	0.855	0.008
	Entropy	0.626	0.628	**0.639**	0.637	0.643	0.648	0.009
	Choquet Integral	0.962	0.966	**0.968**	0.968	0.971	0.975	0.004
	Distance (1)	0.812	0.813	0.821	0.822	0.829	0.836	0.009
	Distance (2)	0.817	0.819	0.826	0.825	0.828	0.833	0.006
	Distance (3)	0.805	0.811	0.817	0.816	0.820	0.828	0.007

Table 3
Experimental results of coverage ratios on four elements of Algorithm 4.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
$\bar{l} = 100, t = 2$	Orness	0.878	0.900	**0.907**	0.910	0.925	0.947	0.021
	Entropy	0.491	0.494	0.496	0.495	0.497	0.498	0.002
	Choquet Integral	0.946	0.950	**0.958**	0.956	0.961	0.962	0.006
	Distance (1)	0.780	0.791	0.808	0.812	0.822	0.873	0.027
	Distance (2)	0.779	0.791	0.814	0.812	0.836	0.843	0.026
	Distance (3)	0.785	0.801	0.805	0.810	0.819	0.839	0.017
$\bar{l} = 100, t = 3$	Orness	0.795	0.829	**0.837**	0.835	0.843	0.855	0.018
	Entropy	0.623	0.627	**0.629**	0.632	0.637	0.649	0.008
	Choquet Integral	0.897	0.929	**0.942**	0.939	0.953	0.961	0.019
	Distance (1)	0.835	0.870	0.879	0.884	0.904	0.929	0.029
	Distance (2)	0.821	0.854	0.872	0.870	0.882	0.908	0.027
	Distance (3)	0.858	0.877	**0.899**	0.895	0.913	0.937	0.026
$\bar{l} = 150, t = 2$	Orness	0.866	0.890	**0.895**	0.896	0.907	0.914	0.015
	Entropy	0.494	0.495	0.495	0.495	0.496	0.497	0.001
	Choquet Integral	0.942	0.952	0.956	0.955	0.960	0.970	0.008
	Distance (1)	0.804	0.823	0.836	0.835	0.847	0.863	0.018
	Distance (2)	0.790	0.806	0.823	0.818	0.825	0.850	0.018
	Distance (3)	0.804	0.825	0.837	0.834	0.845	0.849	0.015
$\bar{l} = 150, t = 3$	Orness	0.800	0.807	**0.809**	0.813	0.815	0.841	0.013
	Entropy	0.612	0.623	**0.627**	0.627	0.629	0.650	0.010
	Choquet Integral	0.929	0.939	**0.943**	0.941	0.945	0.950	0.007
	Distance (1)	0.865	0.882	**0.899**	0.897	0.912	0.924	0.021
	Distance (2)	0.859	0.887	**0.902**	0.896	0.909	0.913	0.018
	Distance (3)	0.860	0.890	**0.895**	0.892	0.898	0.911	0.015

Table 4
Scores of 7 alternatives on 5 criteria.

	1	2	3	4	5
a	0.90	0.55	0.55	0.55	0.90
b	0.90	0.55	0.90	0.55	0.55
c	0.55	0.55	0.90	0.55	0.90
d	0.90	0.90	0.55	0.55	0.55
e	0.55	0.55	0.90	0.90	0.55
f	0.55	0.55	0.90	0.55	0.55
g	0.55	0.55	0.55	0.55	0.90

benchmarks, while the $\bar{l} = 100, t = 2$ case demonstrates two surpassing values, and the $\bar{l} = 150, t = 2$ case has one surpassing value, as indicated in bold font.

5.2. Random generation with additional linear constraints

In this subsection, we adapt the example in [15] to demonstrate the coverage of our random generation methods. Suppose 7 alternatives have partial evaluations on five criteria $N = \{1, 2, 3, 4, 5\}$, as shown in Table 4.

Table 5
Experimental results of coverage ratios with additional constraints.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
Algorithm 2	Deviations	0.910	0.930	**0.940**	0.939	0.947	0.970	0.016
	Split	0.270	0.290	**0.315**	0.333	0.373	0.430	0.055
	Sum	0.930	0.940	**0.960**	0.959	0.978	0.990	0.021
	Distance (1)	0.870	0.880	**0.890**	0.894	0.898	0.940	0.021
	Distance (2)	0.720	0.752	**0.765**	0.760	0.770	0.780	0.017
	Distance (3)	0.850	0.885	**0.900**	0.897	0.918	0.920	0.024
Algorithm 3, $t = 3$	Deviations	0.940	0.952	**0.960**	0.961	0.970	0.980	0.012
	Split	0.180	0.217	0.245	0.244	0.275	0.290	0.036
	Sum	0.920	0.920	0.920	0.923	0.927	0.930	0.005
	Distance (1)	0.610	0.610	0.620	0.627	0.628	0.700	0.028
	Distance (2)	0.460	0.480	0.480	0.489	0.497	0.530	0.022
	Distance (3)	0.630	0.660	0.680	0.670	0.680	0.690	0.018
Algorithm 3, $t = 5$	Deviations	0.910	0.940	**0.940**	0.940	0.950	0.950	0.012
	Split	0.460	0.470	**0.480**	0.487	0.497	0.530	0.023
	Sum	0.880	0.883	0.890	0.891	0.900	0.900	0.009
	Distance (1)	0.490	0.490	0.500	0.508	0.525	0.550	0.021
	Distance (2)	0.370	0.390	0.390	0.396	0.407	0.430	0.017
	Distance (3)	0.550	0.560	0.570	0.573	0.580	0.600	0.017
Algorithm 4, $\bar{l} = 150, t = 3$	Deviations	0.770	0.782	0.795	0.799	0.810	0.840	0.021
	Split	0.070	0.092	0.115	0.117	0.148	0.150	0.030
	Sum	0.910	0.950	**0.950**	0.951	0.968	0.980	0.022
	Distance (1)	0.780	0.812	0.830	0.833	0.858	0.880	0.032
	Distance (2)	0.510	0.530	0.545	0.548	0.560	0.600	0.028
	Distance (3)	0.630	0.670	0.695	0.705	0.737	0.800	0.052
Algorithm 4, $\bar{l} = 150, t = 5$	Deviations	0.720	0.763	0.770	0.772	0.788	0.810	0.028
	Split	0.180	0.227	0.255	0.253	0.288	0.300	0.039
	Sum	0.910	0.923	**0.940**	0.938	0.950	0.970	0.020
	Distance (1)	0.770	0.805	0.825	0.819	0.830	0.850	0.023
	Distance (2)	0.510	0.545	0.575	0.564	0.588	0.590	0.028
	Distance (3)	0.660	0.672	0.700	0.703	0.720	0.770	0.035

Some additional constraints about the importances and interactions among criteria are given as:

$$\begin{aligned} & -0.01 \leqslant I_\mu(\{1\}) - I_\mu(\{2\}), \leqslant I_\mu(\{3\}) - I_\mu(\{4\}) \leqslant 0.01, \\ & I_\mu(\{1,2\}), I_\mu(\{3,4\}) \leqslant -0.05, \\ & I_\mu(\{1,3\}), I_\mu(\{1,4\}), I_\mu(\{1,5\}), I_\mu(\{2,3\}), I_\mu(\{2,4\}), \\ & I_\mu(\{2,5\}), I_\mu(\{3,4\}), I_\mu(\{4,5\}) \geqslant 0.05. \end{aligned} \tag{17}$$

Let's further consider the utilization of 3-additive measures:

$$m_\mu(A) = 0, |A| > 3, A \subseteq N. \tag{18}$$

In this case, we can analyze the coverage based on the objective function indices of identification models. Specifically, for the minimum deviation method, we need the expected overall evaluations of 7 alternatives, which are as follows: 0.697, 0.672, 0.647, 0.622, 0.597, 0.572, and 0.55. For the maximum split methods, we need the expected order of 7 alternatives which can be given as: $a \succ b \succ c \succ d \succ e \succ f \succ g$. Hence we can set up the following constraints:

$$\begin{aligned} & C_\mu(a) - C_\mu(b) \geqslant 0.001, \\ & C_\mu(b) - C_\mu(c) \geqslant 0.001, \\ & \qquad \cdots \\ & C_\mu(f) - C_\mu(g) \geqslant 0.001. \end{aligned} \tag{19}$$

By combining boundary and monotonicity constraints on N, along with the additional conditions specified in Eqs. (17), (18) and (19), we conducted 10 independent runs of Algorithms 2, 3, and 4 respectively to generate 1000 fuzzy measures.

Table 5 shows the statistical information of the coverage ratios about 100 breaks obtained from the different objective functions. According to Eqs. (12), (13), (14) and (15), we can have the range of deviation between Choquet integrals and expected values is $[0.009, 0.222]$, the range of split among 7 alternatives is $[0.001, 0.023]$, the ranges of the sum of Choquet integrals of 7 alternatives are $[3.871, 4.701]$, and the ranges of selected distances (1), (2) and (3) are $[0.000, 0.349]$, $[0.000, 0.791]$ and $[0.006, 0.935]$, respectively. The coverage ratios are calculated using these indices with 100 breaks.

Table 6
Enhancement by increasing the number of generated fuzzy measures.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
Algorithm 4, $\bar{l} = 150, t = 5$ 2000 measures	Deviations	0.710	0.770	0.785	0.781	0.800	0.840	0.037
	Split	0.260	0.293	0.310	0.304	0.320	0.340	0.025
	Sum	0.880	0.930	0.945	0.935	0.958	0.960	0.029
	Distance (1)	0.810	0.820	0.835	0.832	0.840	0.860	0.015
	Distance (2)	0.530	0.560	0.580	0.575	0.590	0.610	0.025
	Distance (3)	0.670	0.683	0.710	0.717	0.755	0.770	0.038

Table 7
Enhancement through convex combination of generated fuzzy measures.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
Algorithm 3, $t = 3$, $t = 3$	Deviations	0.970	0.970	**0.980**	0.977	0.980	0.990	0.007
	Split	0.510	0.532	**0.545**	0.551	0.570	0.600	0.028
	Sum	0.910	0.920	0.920	0.923	0.930	0.930	0.007
	Distance (1)	0.580	0.620	0.635	0.630	0.648	0.660	0.026
	Distance (2)	0.460	0.480	0.485	0.484	0.490	0.500	0.013
	Distance (3)	0.670	0.672	0.680	0.678	0.680	0.690	0.006
Algorithm 3, $t = 5$, $t = 5$	Deviations	0.920	0.940	**0.960**	0.956	0.970	0.980	0.020
	Split	0.540	0.560	**0.580**	0.577	0.595	0.610	0.025
	Sum	0.860	0.883	0.890	0.889	0.898	0.910	0.016
	Distance (1)	0.470	0.492	0.510	0.506	0.525	0.530	0.021
	Distance (2)	0.360	0.383	0.390	0.398	0.410	0.450	0.027
	Distance (3)	0.560	0.570	0.580	0.577	0.580	0.590	0.009

Similarly, we use the medians of the coverage ratios obtained from Algorithm 2 as benchmarks. For Algorithm 3, the case with $t = 3$ achieves a coverage ratio of 0.960, surpassing the benchmark in the deviation index. In addition, the case with $t = 5$ matches the benchmark with a coverage ratio of 0.940 in the deviation index and outperforms in the split index with a coverage ratio of 0.480. For Algorithm 4, the cases with $\bar{l} = 150$ and $t = 3$, as well as $t = 5$, have similar coverage ratios of 0.950 and 0.940, respectively, in the sum index.

The coverage ratios presented in the above tables indicate that while some ratios exceed 0.90, there are still ratios below 0.5. It is worth noting that no single algorithm consistently outperforms others in all indices. This highlights the importance of analyzing and improving the coverage ratio after generating fuzzy measures, which will be the main focus of the upcoming section.

6. Further enhancement of coverage ratio

The low coverage values may be a consequence of the inherent characteristics of the index being used. For example, the split index mentioned in Eq. (13) can be represented as:

$$\epsilon = \min(C_\mu(a) - C_\mu(b), C_\mu(b) - C_\mu(c), \cdots, C_\mu(f) - C_\mu(g)), \quad (20)$$

where the min function highlights the smallest difference among Choquet integrals, which can result in an uneven distribution of the output values. Moreover, even for a linear index such as the distance of a fuzzy measure to a given boundary, as shown in Eq. (16), there can still be an uneven possibility in obtaining its values due to the irregularity of the feasible domain. These uneven distributions make it more challenging to achieve certain index values, ultimately resulting in low coverage ratios.

Another factor contributing to the low coverage ratios may be the contradiction between the random generation methods and the indices used for evaluation. For example, the extreme and vertex measures often have small entropy values close to zero. This poses a challenge for algorithms such as 3 and 4 to generate fuzzy measures with larger entropy values, see e.g., the results in Tables 2 and 3.

Once we obtain low coverage ratios, there are several approaches available to enhance these index values. A direct approach to enhance the coverage ratios is to increase the number of generated fuzzy measures. For instance, by doubling the generation number of Algorithm 4 for $\bar{l} = 150, t = 5$ to 2000 fuzzy measures, we observe an improvement in the coverage ratios, see Table 6 and the last section of Table 5.

Another approach to generate additional fuzzy measures is through the use of the convex combination method described in Eq. (5). Table 7 presents the results of applying this method to further generate 1000 fuzzy measures. Comparing these results with the second and third sections of Table 5, we can observe the improvements in the coverage ratios.

The third approach is to combine the results of different algorithms. Table 8 demonstrates three combinations of Algorithms 2, 3, and 4. It can be seen that the coverage ratio, as indicated by the sum of Choquet integrals, can reach 1, and several values outperform the benchmarks.

The final approach involves generating additional fuzzy measures to fill the unoccupied breaks of index values using optimization models. This approach is specifically suitable for linear index-based coverage ratios. For example, if the orness index has an

Table 8
Enhancement by collecting results from different algorithms.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
Algorithm 3, $t = 3$	Deviations	0.960	0.960	**0.960**	0.964	0.970	0.970	0.005
plus with	Split	0.220	0.275	0.300	0.286	0.300	0.320	0.031
Algorithm 4, $\bar{l} = 150, t = 3$	Sum	1.000	1.000	**1.000**	1.000	1.000	1.000	0.000
	Distance (1)	0.790	0.805	0.830	0.826	0.840	0.860	0.023
	Distance (2)	0.630	0.640	0.650	0.656	0.660	0.720	0.025
	Distance (3)	0.830	0.840	0.845	0.846	0.850	0.870	0.013
Algorithm 3, $t = 5$	Deviations	0.940	0.950	**0.960**	0.960	0.968	0.980	0.013
plus with	Split	0.440	0.470	**0.480**	0.481	0.497	0.510	0.023
Algorithm 4, $\bar{l} = 150, t = 5$	Sum	0.990	1.000	**1.000**	0.998	1.000	1.000	0.004
	Distance (1)	0.800	0.830	0.835	0.843	0.858	0.890	0.028
	Distance (2)	0.560	0.562	0.590	0.596	0.608	0.660	0.028
	Distance (3)	0.760	0.790	0.795	0.797	0.815	0.820	0.019
Algorithm 2	Deviations	0.940	0.960	**0.960**	0.961	0.970	0.970	0.010
plus with	Split	0.430	0.447	**0.485**	0.481	0.500	0.560	0.039
Algorithm 3, $t = 3$	Sum	0.970	0.973	**0.980**	0.982	0.987	1.000	0.011
	Distance (1)	0.850	0.880	**0.890**	0.894	0.922	0.940	0.032
	Distance (2)	0.730	0.772	**0.785**	0.779	0.790	0.800	0.021
	Distance (3)	0.860	0.875	**0.900**	0.898	0.920	0.930	0.025

Table 9
Enhancement by makeup the unoccupied breaks of indices for Algorithm 1.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
Orness Makeup	Orness	1.000	1.000	**1.000**	1.000	1.000	1.000	0.000
	Entropy	0.903	0.906	**0.908**	0.909	0.913	0.914	0.004
	Choquet Integral	0.950	0.954	**0.954**	0.957	0.960	0.968	0.005
	Distance (1)	0.893	0.902	**0.903**	0.904	0.909	0.913	0.006
	Distance (2)	0.894	0.898	**0.905**	0.906	0.913	0.919	0.009
	Distance (3)	0.895	0.897	**0.901**	0.904	0.912	0.919	0.009
Choquet integral Makeup	Orness	0.837	0.846	**0.849**	0.850	0.857	0.859	0.007
	Entropy	0.874	0.882	**0.888**	0.885	0.889	0.893	0.007
	Choquet Integral	1.000	1.000	**1.000**	1.000	1.000	1.000	0.000
	Distance (1)	0.878	0.893	**0.897**	0.895	0.899	0.906	0.009
	Distance (2)	0.887	0.892	**0.899**	0.898	0.904	0.908	0.007
	Distance (3)	0.881	0.891	**0.898**	0.894	0.900	0.901	0.007

unoccupied break $[a, b]$, we can use the following model to generate a fuzzy measure with an orness value equal to a random value $c \in [a, b]$:

$$\begin{aligned} &\min z = d^+ + d^- \\ &\text{s.t. } \mathcal{B} \text{ and } \mathcal{D}, \\ &orness(\mu) - d^+ + d^- = c. \end{aligned} \tag{21}$$

Due to the convexity of the feasible domain, the optimal objective function in the above model should always be 0. Table 9 demonstrates the enhancements achieved through orness and Choquet integral for Algorithm 1 in generating fuzzy measures for $N = \{1, 2, 3, 4\}$ under normalization and monotonicity conditions, providing a comparison with the results from Table 1.

With these random generation algorithms and coverage ratio improvement approaches, we can investigate some applications in the next section.

7. Possible applications of random generation methods

First, let's consider the optimization of a non-linear function known as the Rastrigin function, represented as:

$$f(\mu) = 20\,2^n + \sum_{A \subseteq N} \left(\mu(A)^2 - 20\cos(2\pi\mu(A))\right),$$

where $N = \{1, 2, 3, 4\}$. The Rastrigin function is commonly used as a benchmark function in optimization algorithms and is known for its multimodal and highly rugged landscape, with many local minima and a single global minimum at the points where all variables are zero.

Table 10
Results of optimization solutions of Rastrigin function.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
1000 measures by Algorithm 2	Maximum	227.755	231.292	235.813	237.876	241.283	260.130	9.370
	Minimum	8.131	9.013	10.548	11.768	14.291	18.333	3.593
5000 measures by Algorithm 2	Maximum	240.366	245.356	249.561	250.211	252.406	265.398	7.223
	Minimum	5.570	8.118	8.463	8.229	8.820	9.232	1.030
5000 measures by Algorithm 2	Maximum	244.518	250.312	251.068	253.538	257.108	271.229	8.075
and orness based improvement	Minimum	1.000	1.000	1.000	1.000	1.000	1.001	0.000

Table 11
Results of machine learning using Minkowski distances.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
$p = 1$	Maximum deviation	0.486	0.486	0.486	0.486	0.486	0.486	0.000
	Minimum deviation	0.024	0.034	0.038	0.036	0.040	0.041	0.005
$p = 2$	Maximum deviation	0.222	0.222	0.222	0.222	0.222	0.222	0.000
	Minimum deviation	0.011	0.017	0.021	0.020	0.022	0.024	0.004
$p = 3$	Maximum deviation	0.178	0.178	0.178	0.178	0.178	0.178	0.000
	Minimum deviation	0.008	0.015	0.017	0.016	0.018	0.020	0.003

Table 12
Different orderings of 7 alternatives and their frequencies.

No.	Ordering	Frequency
1	$e \prec f \prec g \prec c \prec d \prec a \prec b$	108/1062
2	$d \prec g \prec a \prec e \prec f \prec c \prec b$	72/1062
3	$g \prec d \prec a \prec e \prec f \prec c \prec b$	56/1062
4	$a \prec c \prec d \prec e \prec f \prec g \prec b$	46/1062
5	$d \prec g \prec a \prec f \prec e \prec c \prec b$	45/1062
6	$g \prec d \prec a \prec f \prec e \prec c \prec b$	44/1062
7	$d \prec e \prec f \prec b \prec g \prec a \prec c$	30/1062
8	$e \prec f \prec g \prec c \prec d \prec b \prec a$	29/1062
9	$d \prec g \prec a \prec f \prec e \prec b \prec c$	20/1062
10	$g \prec e \prec f \prec c \prec d \prec a \prec b$	25/1062

By using Algorithm 2 and orness based coverage ratio improvement approach 10 times, we have the following results of minimum of maximum values in Table 10. One can figure out that the combination of generation of 5000 measures and orness based coverage ration improvement approach can reach the global minimum value of 1 with the solution as: $\mu(N) = 1$ and $\mu(A) = 0, A \subset N$.

Second, let's examine an application in machine learning. Specifically, in subsection 5.2, we address the task of measuring the deviation between Choquet integrals and the expected values of 7 alternatives, which is a typical machine learning problem. The deviation as shown in Eq. (12) is basically the euclidean distance or a type of Minkowski distance:

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}},$$

where p is a positive parameter, set to 1 for Manhattan distance and 2 for Euclidean distance. Table 11 shows the deviations obtained by using Minkowski distances with $p = 1, 2, 3$ with Algorithm 3 ($t = 3$) and orness based coverage ratio improvement approach.

Third, we investigate the decision analysis situation. Still the example in subsection 5.2, if remove the constraint Eq. (19) and only consider boundary and monotonicity constraints and Eqs. (17) and (18), we have the different ordering of 7 alternatives, Table 12 shows the top 10 relationships and their frequencies by using Algorithm 3 ($t = 3$) and orness based coverage ratio improvement approach (total 1062 fuzzy measures).

Finally, we provide a small example to illustrate a property investigation problem. We focus on the interaction indices of the 1062 fuzzy measures mentioned earlier and specifically investigate the indices of subsets consisting of three criteria. It is worth noting that subsets of other cardinalities are all constrained, as indicated in Eqs. (17) and (18). In Table 13, we present the interaction indices of all subsets with three criteria. Interestingly, we observe that subsets $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{2, 3, 4\}$ and $\{3, 4, 5\}$ all exhibit negative interaction values, even without any explicit constraints on them. This observation prompts further investigations and analyses of their properties, which can be explored by establishing optimization models.

Table 13
Results of interaction indices of subsets with three criteria.

Subsets	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	sd.
$\{1,2,3\}$	-0.570	-0.104	-0.081	-0.089	-0.048	**0.000**	0.071
$\{1,2,4\}$	-0.570	-0.100	-0.079	-0.091	-0.050	**0.000**	0.072
$\{1,2,5\}$	-0.680	-0.100	-0.062	-0.087	-0.026	0.113	0.101
$\{1,3,4\}$	-0.315	-0.104	-0.076	-0.065	-0.042	0.570	0.089
$\{1,3,5\}$	-0.315	-0.027	0.032	0.056	0.128	0.490	0.130
$\{1,4,5\}$	-0.315	0.005	0.061	0.080	0.155	0.528	0.130
$\{2,3,4\}$	-0.670	-0.111	-0.082	-0.100	-0.051	**0.000**	0.084
$\{2,3,5\}$	-0.255	0.015	0.078	0.103	0.182	0.528	0.124
$\{2,4,5\}$	-0.315	-0.017	0.037	0.073	0.157	0.528	0.135
$\{3,4,5\}$	-0.680	-0.149	-0.071	-0.116	-0.032	**0.000**	0.128

8. Conclusions

In this research, we have introduced three distinct approaches, namely random-node based topological sorting, convex combination of constrained extreme measures, and vertex measures, to facilitate the random generation of fuzzy measures under complex linear constraints. By evaluating coverage ratios, which encompass fuzzy measure indices, objective functions, and distances to boundaries, we have empirically demonstrated the effectiveness of these random generation methods in exploring the convex feasible domain. Furthermore, we have improved the coverage performance of these methods by incorporating four different coverage ratio improvement approaches. The integration of random generation techniques with these enhancement strategies has the potential to find applications in diverse domains, such as optimization solutions, machine learning, and multiple criteria decision analyses. The upcoming research tasks may prioritize the practical implementation of these methods within specific optimization and decision analysis contexts. Furthermore, additional investigation is required to explore the adaptation of these techniques and approaches for addressing nonlinear and convex situations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

The work was supported by the Australian Research Council Discovery project DP210100227.

References

[1] G. Beliakov, Construction of aggregation functions from data using linear programming, Fuzzy Sets Syst. 160 (2009) 65–75.
[2] G. Beliakov, On random generation of supermodular capacities, IEEE Trans. Fuzzy Syst. 30 (2022) 293–296.
[3] G. Beliakov, F.J. Cabrerizo, E. Herrera-Viedma, J.Z. Wu, Random generation of k-interactive capacities, Fuzzy Sets Syst. 430 (2022) 48–55.
[4] G. Beliakov, S. James, J.Z. Wu, Discrete Fuzzy Measures: Computational Aspects, Springer, Cham, Switzerland, 2019.
[5] G. Beliakov, J.Z. Wu, Learning fuzzy measures from data: simplifications and optimisation strategies, Inf. Sci. 494 (2019) 100–113.
[6] T. Calvo, B. De Baets, Aggregation operators defined by k-order additive/maxitive fuzzy measures, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6 (1998) 533–550.
[7] A. Chateauneuf, J.Y. Jaffray, Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion, Math. Soc. Sci. 17 (1989) 263–283.
[8] G. Choquet, Theory of capacities, Ann. Inst. Fourier 5 (1954) 131–295.
[9] E. Combarro, P. Miranda, Identification of fuzzy measures from sample data with genetic algorithms, Comput. Oper. Res. 33 (2006) 3046–3066.
[10] E.F. Combarro, I. Díaz, P. Miranda, On random generation of fuzzy measures, Fuzzy Sets Syst. 228 (2013) 64–77.
[11] E.F. Combarro, J.H. de Saracho, I. Díaz, Minimals plus: an improved algorithm for the random generation of linear extensions of partially ordered sets, Inf. Sci. 501 (2019) 50–67.
[12] J. Dujmovic, Two integrals related to means, Univ. Beogr. Publ. Elektroteh. Fak. (1973) 231–232.
[13] M. Grabisch, k-Order additive discrete fuzzy measures and their representation, Fuzzy Sets Syst. 92 (1997) 167–189.
[14] M. Grabisch, Set Functions, Games and Capacities in Decision Making, Springer, Berlin, New York, 2016.
[15] M. Grabisch, I. Kojadinovic, P. Meyer, A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: applications of the Kappalab R package, Eur. J. Oper. Res. 186 (2008) 766–785.
[16] M. Grabisch, C. Labreuche, A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid, Ann. Oper. Res. 175 (2010) 247–286.
[17] M. Grabisch, M. Roubens, Probabilistic interactions among players of a cooperative game, in: M. Machina, B. Munier (Eds.), Beliefs, Interactions and Preferences in Decision Making, Springer, Heidelberg, New York, 1999, pp. 205–216.
[18] T.C. Havens, A.J. Pinar, Generating random fuzzy (capacity) measures for data fusion simulations, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.

[19] I. Kojadinovic, Minimum variance capacity identification, Eur. J. Oper. Res. 177 (2007) 498–514.
[20] J.L. Marichal, Entropy of discrete Choquet capacities, Eur. J. Oper. Res. 137 (2002) 612–624.
[21] J.L. Marichal, k-Intolerant capacities and Choquet integrals, Eur. J. Oper. Res. 177 (2007) 1453–1468.
[22] J.L. Marichal, M. Roubens, Determination of weights of interacting criteria from a reference set, Eur. J. Oper. Res. 124 (2000) 641–650.
[23] R. Mesiar, A. Kolesárová, k-Maxitive aggregation functions, Fuzzy Sets Syst. 346 (2018) 127–137.
[24] P. Meyer, M. Roubens, Choice, ranking and sorting in fuzzy multiple criteria decision aid, in: J. Figueira, S. Greco, M. Ehrogott (Eds.), Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, New York, 2005, pp. 471–503.
[25] P. Miranda, M. Grabisch, P. Gil, p-Symmetric fuzzy measures, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (2002) 105–123.
[26] T. Murofushi, M. Sugeno, An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure, Fuzzy Sets Syst. 29 (1989) 201–227.
[27] M. Sugeno, Theory of Fuzzy Integrals and Its Applications, Ph.D. thesis, Tokyo Institute of Technology, 1974.
[28] Z. Wang, G.J. Klir, Generalized Measure Theory, Springer Science & Business Media, New York, 2010.
[29] J.Z. Wu, G. Beliakov, Nonadditivity index and capacity identification method in the context of multicriteria decision making, Inf. Sci. 467 (2018) 398–406.
[30] J.Z. Wu, G. Beliakov, Nonadditive robust ordinal regression with nonadditivity index and multiple goal linear programming, Int. J. Intell. Syst. 34 (2019) 1732–1752.
[31] J.Z. Wu, L. Huang, R.J. Xi, Y.P. Zhou, Multiple goal linear programming-based decision preference inconsistency recognition and adjustment strategies, Information 10 (2019) 223.
[32] J.Z. Wu, Q. Zhang, Q. Du, Z. Dong, Compromise principle based methods of identifying capacities in the framework of multicriteria decision analysis, Fuzzy Sets Syst. 246 (2014) 91–106.