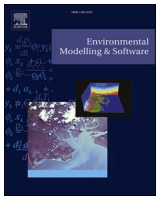
Environmental Modelling and Software 148 (2022) 105282

Contents lists available at ScienceDirect

Environmental Modelling and Software

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



The pie sharing problem: Unbiased sampling of N+1 summative weights☆

Juliane Mai *, James R. Craig, Bryan A. Tolson

Civil and Environmental Engineering, University of Waterloo, 200 University Ave West, Waterloo, ON, N2L 3G1, Canada

ARTICLE INFO

Keywords:
Weight sampling
Identical distributions
Random number generation
Constrained random numbers

ABSTRACT

A simple algorithm is provided for randomly sampling a set of N+1 weights such that their sum is constrained to be equal to one, analogous to randomly subdividing a pie into N+1 slices where the probability distribution of slice volumes are identically distributed. The cumulative density and probability density functions of the random weights are provided. The algorithmic implementation for the random number sampling are made available. This algorithm has potential applications in calibration, uncertainty analysis, and sensitivity analysis of environmental models. Three example applications are provided to demonstrate the efficiency and superiority of the proposed method compared to alternative sampling methods.

1. Introduction

There are many scenarios in optimization, sensitivity analysis, and uncertainty analysis of environmental models where it is desirable to randomly sample $N + 1$ quantities w_i such that the sum of quantities is constrained to be equal to a constant, here set to 1 for simplicity:

$$\sum_{i=1}^{N+1} w_i = 1 \quad . \tag{1}$$

This problem is equivalent to the problem of sampling N randomly sized slices of apple pie; one slice for each of your friends while the remaining last slice ($N + 1$) is for yourself. We assume that you do not favour any of your friends (i.e., sampling distribution of all slices is identical), the order you hand out the slices does not matter (the samples are independent) and you will eat all leftovers yourself (all slices sum up to one).

Examples from environmental modelling include applications where one wants to randomly partition unity into classes where class membership is assumed to have equal likelihood. For instance, one may wish to randomly sample the soil texture triangle which characterizes soils by percent silt, sand, or clay (Pozdniakov et al., 2019). In ecological modelling, one may wish to randomly generate distributions of biomass compositions by percent lipids, carbohydrates, and proteins. Summative weights are used in environmental modeling when various alternative options are blended/mixed. These options could be, for example, several precipitation station datasets surrounding a domain of interest. Instead of forcing the model with individual station datasets, one could use the weighted average of the datasets (Piotrowski et al., 2019; Montanari and Di Baldassarre, 2013). Weights also need to be sampled in multi-model averaging approaches such as in Arsenault et al. (2015) or to weight multiple performance criteria in decision making applications (Ganji et al., 2016; Hyde and Maier, 2006; Hyde et al., 2004). Another example are so-called blended models where the model is defined as the weighted average of multiple models (Mai et al., 2020). To conserve mass or otherwise respect constraints in these examples, the weights are required to sum up to 1. Many optimization algorithms similarly depend upon uniform sampling of parameter space – when model parameters are constrained as expressed in equation (1), it is useful to have the ability to sample independently and identically while respecting constraints upon parameters.

While it is straightforward to sample weights for $N = 1$ by sampling from a uniform distribution and then determining w_2 as $1 - w_1$, this simple approach is not easily extendable to N greater than 1. Two naïve approaches are, for example, (a) to sample random numbers iteratively by reducing the sampling range in each step such that the overall sum can not be exceeded (see Section 4 for algorithmic details), or (b) drawing N samples from the uniform distribution and then either discard the entire sample in case the sum is already exceeding 1 otherwise keeping the sample and assigning the remainder to 1 to the $(N + 1)^{\text{th}}$ weight. These approaches, however, can introduce a bias in the expected values of weights when using approach (a), i.e., $w = \{0.25, 0.25, 0.25, 0.25\}$ is more likely to be generated than $w = \{0, 0, 0, 1\}$ for $N = 3$. This bias is due to the fact that a large last weight (here w_4) requires all previous weights to be small while a large first weight does not have that

☆ The fully documented codes and examples are available on GitHub https://github.com/julemai/PieShareDistribution.

* Corresponding author.
E-mail address: juliane.mai@uwaterloo.ca (J. Mai).

https://doi.org/10.1016/j.envsoft.2021.105282
Received 25 June 2021; Received in revised form 24 November 2021; Accepted 7 December 2021
Available online 14 December 2021

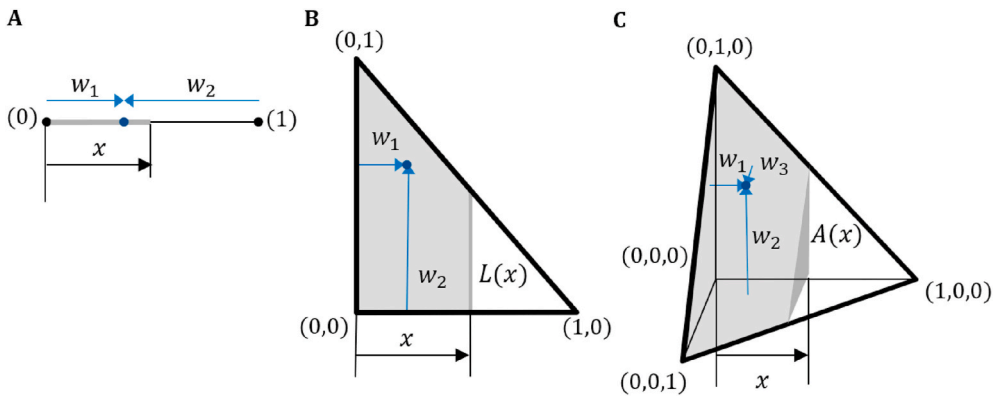


Fig. 1. (A) Sampling weights along a line of length 1. (B) Sampling positions within a triangle with unit length sides, with the sweeping area from $x = 0$ in gray. (C) The sweeping volume from $x = 0$ in a tetrahedron with unit length sides. Note that the visual depiction of the weights w_i and the location of the sweeping areas are independent.

restriction. Naïve approach (b), on the other hand, does not draw biased samples but is significantly more computationally expensive than approach (a) or other methods. These limitations will be demonstrated in detail later in this work (see Section 5 and Figs. 5, 6, and 7 therein).

This challenge of sampling unbiased weights has been addressed in the computational task scheduling literature (Bini and Buttazzo, 2005) and the statistics literature (Moeini et al., 2011). The first algorithmic implementation to this problem we could locate is the UUnifast sampling algorithm first proposed by Bini and Buttazzo (2005). Another subsequent solution is found by Moeini et al. (2011) who do not cite Bini and Buttazzo (2005) and is thus presumably independent of Bini and Buttazzo (2005). A key difference between these solutions is that Bini and Buttazzo (2005) provides a readily available sampling algorithm (UUnifast) while Moeini et al. (2011) only describes the mathematical underpinning of an approach for sampling N random numbers from the uniform distribution, $\mathcal{U}[0,1]$, and transforming them into $N + 1$ weights. As reported in Griffin et al. (2020), an alternate method for sampling from a uniform distribution is available within an open-source Matlab code (Stafford, 2006) but no derivation accompanies the tool. The methodology used is unpublished and not readily transferable to applications outside Matlab; the CDFs, PDFs, and inverse CDFs of the resultant weight distributions are not reported. Griffin et al. (2020) proposed a new and more general sampling algorithm, called the Dirichlet-Rescale algorithm (DRS), that replaces the UUnifast algorithm (Bini and Buttazzo, 2005) and can sample a vector of $N + 1$ outputs that sum to U (not just 1), with each individual output value in the range of potentially different upper and lower bounds. The mathematical presentation in these publications is quite advanced and as noted by Bini and Buttazzo (2005), there are some hidden pitfalls with the implementation of such a sampling algorithm. These pitfalls include the biased sampling of the weights (some weight combinations are more likely to appear than others) and the poor computational efficiency of some implementations (Bini and Buttazzo, 2005). None of the aforementioned methods has been published in an environmental modelling related journal which has apparently hampered their practical application in this field: for example, none of the 867 publications that cite Bini and Buttazzo (2005) (status: June 2021) described applications in the environmental sciences.

The earlier mentioned examples of environmental modelling publications that require the sampling of weights (Pozdniakov et al., 2019; Piotrowski et al., 2019; Montanari and Di Baldassarre, 2013; Arsenault et al., 2015; Ganji et al., 2016; Hyde and Maier, 2006; Hyde et al., 2004) do not describe their sampling method explicitly and do not cite the methods introduced by Bini and Buttazzo (2005), Moeini et al. (2011), Stafford (2006), or Griffin et al. (2020). As such, some of these previous studies might not sample correctly.

The work presented here provides an alternate (independently obtained) mathematical derivation of the approach of Bini and Buttazzo (2005) and Moeini et al. (2011). It presents the full derivation of PDFs and CDFs and outlines a straightforward and replicable algorithm for determining unbiased weights which satisfy equation (1). The method has recently been used by Mai et al. (2020) to enable the estimation of model structural sensitivities through the weighting of discrete model process options continuously for simulating process-level hydrologic fluxes. The method also found its application in Chlumsky et al. (2021) to allow for a simultaneous calibration of model parameters and model structure where the weights are used to define the latter.

This work aims to bring the proper sampling of weights to the attention of environmental modellers by providing not only the mathematical underpinning (Section 2), the resulting explicit method to sample the weights (Section 3) and algorithmic validity checks of the proposed sampling method (Section 4) but also by presenting a few example applications requiring the sampling of weights (Section 5). The algorithms for the weight sampling are provided on GitHub in multiple programming languages such that they can easily be used or translated into other programming languages.

2. Methodology

For the case of two weights summing to one, we can consider the constrained sampling problem as trying to uniformly sample a position along the one-dimensional line from $x = 0$ to 1, where the distance to the left vertex ($x = 0$) is given by w_1 and the distance to the right vertex ($x = 1$) is given by w_2, as depicted in Fig. 1A. From simple geometry (or equation (1)), $w_2 = 1 - w_1$, by definition. Because we want to uniformly sample the domain, the probability of being at any one position x along the line is equal, i.e., $f(x) = 1$ and, since w_1 is interpreted as the distance from $x = 0$, w_1 is equal to x, and therefore $f(w_1)$ is a uniform distribution over the interval from 0 to 1. Note that the same holds for w_2 as $w_2 = 1 - x = 1 - w_1$, meaning $f(w_2)$ is also a uniform distribution.

For N greater than 1, the same general approach is valid– we can consider the sampling problem as equivalent to uniformly sampling a single position $(w_1, w_2, \ldots, w_{N+1})$ from a regular N-simplex, which is a N-dimensional symmetric geometric shape with N edges connecting N nodes. The simplex we use here is defined as the hypervolume defined by all points $(w_1, w_2, \ldots, w_{N+1})$ where w_i is greater than zero for all i such that the sum of the coordinates is less than or equal to 1. For $N = 1$, the simplex is the line from above; for $N = 2$, the simplex is the right triangle of Fig. 1B; for $N = 3$, the simplex is a tetrahedron, and so on. We wish to uniformly sample points from the hypervolume of each simplex. This is the reasoning for the scaling of the simplex: a vertex corresponds to the case where one weight is equal to one, and all others are zero.

The uniform sampling strategy may be visualized readily for $N = 2$, as shown in Fig. 1B– uniform sampling corresponds to even point density

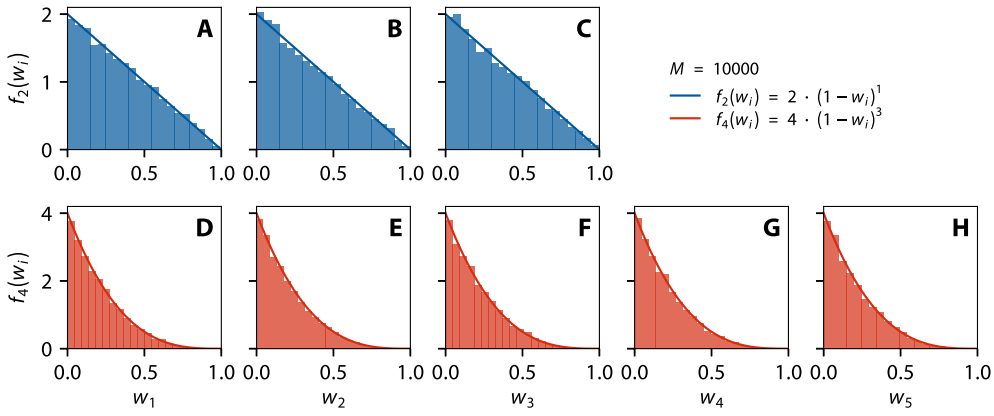


Fig. 2. First row (A–C) shows $N+1=3$ and second row (D–H) shows $N+1=5$ sampled weights (discrete histogram) atop of their according continuous PDF $f_N(w_i)$ (Eq. (9)). The number of samples drawn is $M = 10\,000$ in both cases.

of randomly selected points within this triangle, without position bias. It is worthwhile to note, that for $N = 2$, each position can be uniquely identified using a single 2D coordinate (w_1, w_2), i.e., there are only two degrees of freedom and the final weight may always be defined by knowing the other coordinates, e.g., $w_3 = 1 - w_1 - w_2$; this holds true in higher dimensions as well.

With reference to Fig. 1B, we here note that we can determine a uniform sampling strategy by recognizing that the cumulative distribution function of each weight should be equal to the relative area swept from the edge i to a distance of w_i from the edge of the triangle, regardless which edge it sweeps from, i.e.,

$$F_2(w_i) = \frac{1}{V_2}\int_0^{w_i} (1)\cdot(1-x)\ dx = 1-(1-w_i)^2 \tag{2}$$

where F_2 is the cumulative distribution function and $V_N = 1/N!$ is the hyper-dimensional volume of an N-simplex with unit-length edges, which in this case is the area of the triangle. Here, $L(x) = (1)\cdot(1-x)$ is the length of the sweeping edge expressed as a function of x, where the 1 is used to indicate the length of the edge at $x = 0$. The probability distribution function of w_i is given by the derivative of $F_2(w_i)$,

$$f_2(w_i) = \frac{dF_2}{dw_i} = 2\cdot(1-w_i)\ . \tag{3}$$

This can be repeated for higher dimensions. For example, for $N = 3$,

$$F_3(w_i) = \frac{1}{V_3}\int_0^{w_i} A(x)\ dx \tag{4}$$

where $A(x) = A(0)\cdot(1-x)^2$ is the area of the sweeping face at a distance x from side i, which is a triangle with side lengths that scale with $1-x$ and a maximum area $A(0)$, the area of the surface corresponding to $x = 0$. Note that this area is equivalent to the hypervolume V_2 (note also that the edge $L(0)$ was the same as V_1), giving

$$F_3(w_i) = \frac{V_2}{V_3}\int_0^{w_i} (1-x)^2\ dx = 1-(1-w_i)^3\ . \tag{5}$$

Differentiating with respect to w_i provides the probability density function for w_i:

$$f_3(w_i) = \frac{dF_3}{dw_i} = 3\cdot(1-w_i)^2 \tag{6}$$

Note that the sweeping face was a line of length proportional to $1-x$ in 2D, the area of an equilateral triangle, $A_3(x)$, with side length proportional to $1-x$ in 3D, and will be the volume of a pyramid with side length decreasing as $1-x$ in 4D, i.e., $A_4(x) = A_4(0)\cdot(1-x)^3$. This generalizes to higher dimensions as $A_N(x)/A_N(0) = (1-x)^{N-1}$, leading to the following integral:

$$F_N(w_i) = \frac{V_{N-1}}{V_N}\int_0^{w_i}\frac{A_N(x)}{A_N(0)}\ dx = N\int_0^{w_i}(1-x)^{N-1}\ dx\ . \tag{7}$$

This, in turn, leads to a general expression for both the CDF and PDF of the weights:

$$F_N(w_i) = 1-(1-w_i)^N \tag{8}$$

$$f_N(w_i) = N\cdot(1-w_i)^{N-1}\ . \tag{9}$$

Note that these distributions more heavily lean towards smaller values of w_i as N increases, which has a clear interpretation. For instance, it should be expected that the mean weight of all sampled weights for $N+1=3$ is 1/3 and the mean for $N+1=7$ should be 1/7, i.e., a uniform weighting. This can be shown to be true for any N, i.e.,

$$\bar{w} = \int_0^1 w\cdot f_N(w)\ dw = N\int_0^1 w(1-w)^{N-1}\ dw = \frac{1}{N+1}\ . \tag{10}$$

3. Sampling

Of course, these distribution functions cannot be sampled independently– once w_1 is sampled. For instance, the other weights, w_2, w_3, etc., are constrained to a single plane slicing through the N simplex. The appropriate strategy for such conditional sampling recognizes that, once the first weight is sampled, the determination of the $N-1$ remaining weights may be done by sampling from the distribution $f_{N-1}(w)$, then scaling the weights by $1-w_1$, analogous to sampling the remainder of the pie into N-1 slices. This can be repeated from N down to 1. We can define, from equation (8) above, the following sampling function (or inverse CDF), enabling us to generate a random number sampled from the distribution $f_N(w)$ given a random number r sampled from a uniform distribution in the range from 0 to 1:

$$S_N(r) = 1-(1-r)^{\frac{1}{N}} \tag{11}$$

Our sampling strategy then can be summarized as follows. For each set of weights needed, first generate a vector of random numbers (r_1, r_2, …, r_N) from a uniform distribution between 0 and 1. The corresponding vector of weights (w_1, w_2, …, w_{N+1}) can then be calculated using:

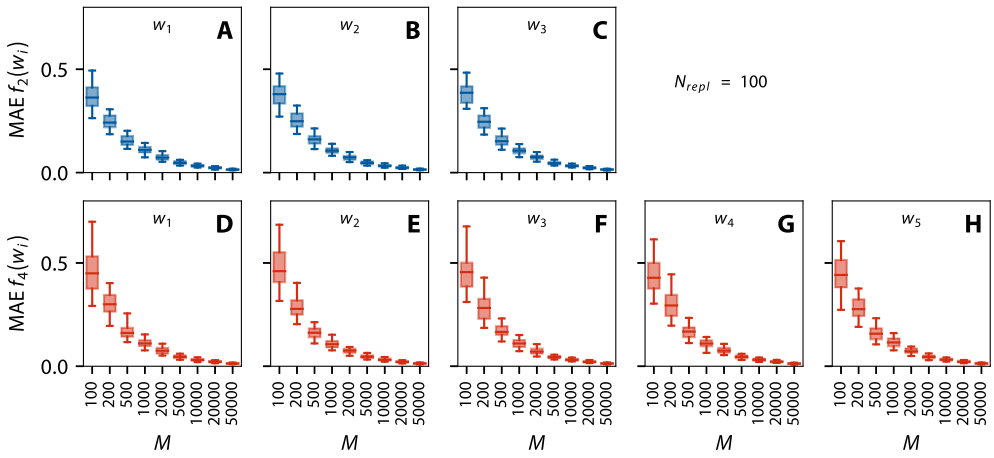


Fig. 3. The mean absolute error (MAE) of the sampled distribution of the weights w_i are shown for $N + 1 = 3$ (A–C) and $N + 1 = 5$ (D–H) weights drawing different numbers of samples M. The boxplots are derived by performing $N_{repl} = 100$ independent sampling experiments where each time M samples of $N + 1$ weights have been drawn and compared to the analytical PDF $f_N(w_i)$ (Eq. (9)).

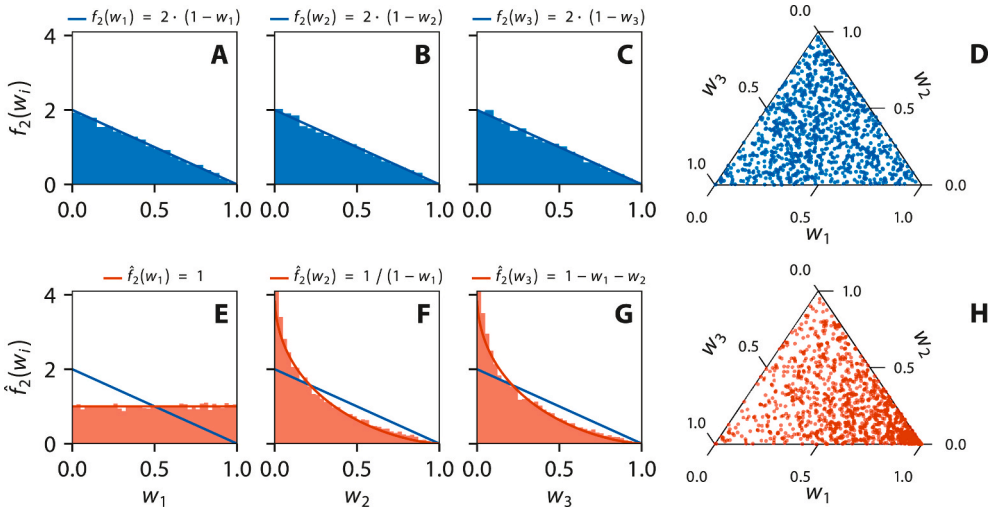


Fig. 4. (A–D) show the results of the proposed sampling for $N + 1 = 3$ weights w_i following the sampling algorithm of equation (12). Panel (D) depicts the uniform distribution of the three weights using a ternary plot. (E–G) shows the comparative results when a naïve sampling is performed by sampling the first weight w_1 uniformly in the unit interval, the second weight w_2 is determined by scaling a second uniform random number within the remaining range $1 - w_1$ and the third weight is the remainder $1 - w_1 - w_2$. The ternary plot (H) shows the biased distribution of the three weights following this naïve approach. The blue lines in panels (E–G) are added as a reference to demonstrate the bias in the sampling of the weights. The number of samples drawn is $M = 10\,000$ in both cases. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

$$
\begin{aligned}
w_1 &= S_N(r_1) \\
w_2 &= (1 - w_1) S_{N-1}(r_2) \\
w_3 &= (1 - w_1 - w_2) S_{N-2}(r_3) \\
&\vdots \\
w_j &= \left(1 - \sum_{i=1}^{j-1} w_i\right) S_{N-j-1}(r_j) \\
&\vdots \\
w_{N+1} &= \left(1 - \sum_{i=1}^{N} w_i\right) .
\end{aligned}
\tag{12}
$$

4. Demonstration of validity of proposed sampling method

A first sampling experiment is performed using $M = 10\,000$ independent samples of weights drawn (Fig. 2). The sampling is tested using $N + 1 = 3$ and $N + 1 = 5$ following the sampling strategy in equation (12). Fig. 2 demonstrates that the histograms of the sampled weights are all identical and matching the analytical PDF $f_2(w_i)$ and $f_4(w_i)$ depending if 3 or 5 weights are sampled.

A second numerical exercise is used to confirm that the approximated and analytical PDF also match when the $N_{repl} = 100$ independent

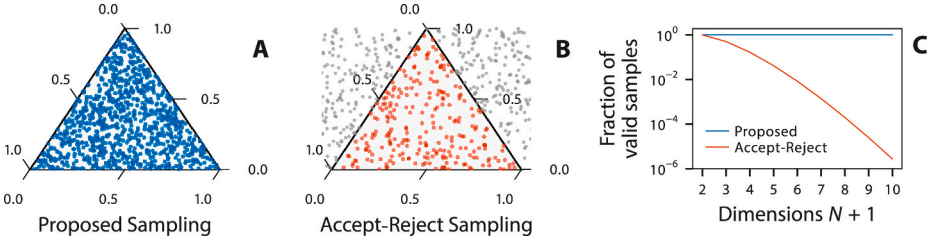


Fig. 5. Two methods to sample $N + 1 = 3$ weights are visualized in a ternary plot: (A) the proposed method and (B) the Accept-Reject sampling where a samples are drawn within a unit hypercube (here: a unit square) and non-feasible points (gray) are withdrawn. (C) Shows that fraction of feasible points. The analytically-derived efficiency of the Accept-Reject sampling is $1/N!$ The proposed sampling is drawing only feasible points (blue line), and hence is constantly 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

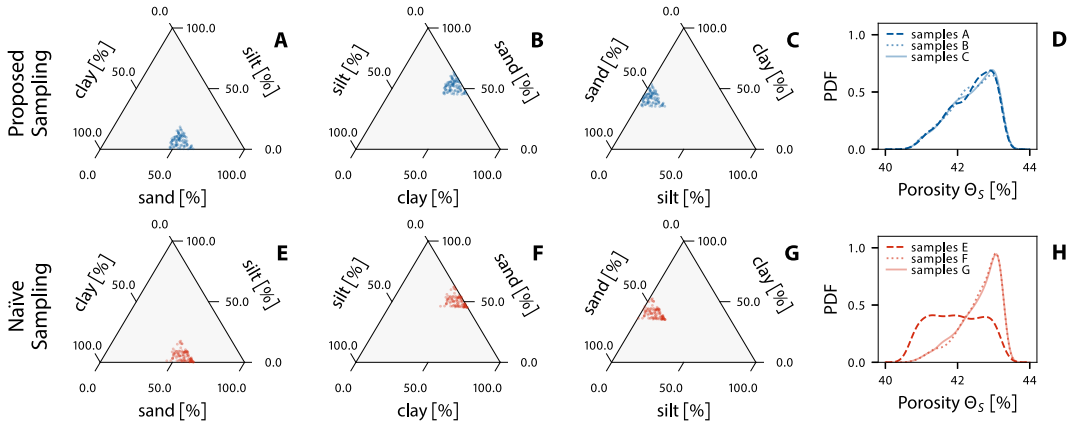


Fig. 6. Sampling of the USDA soil class "sandy clay" using (A–D) the proposed sampling and (E–H) the naïve approach. The first three plots in each row show the same samples; only the assignment of sand, clay and silt content are assigned to different axes. The forth panel in both rows shows the distribution of the derived porosity Θ_S based on the sampled soil textures using the functional relationship by Cosby et al. (1984). Three distributions are shown corresponding to the samples of the first three panels in each row.

samplings are performed while increasing the number of samples M from 100 to 50 000 (Fig. 3). The results show that the mean absolute error between the continuous analytical PDF (Eq. (9)) and the approximated PDF are converging to zero with an increasing number of samples drawn. $M = 10\,000$ proves to be a sufficient number of samples to yield a mean absolute error (MAE) of below 0.06 for all 100 repetitions of the sampling experiment. The uncertainty of the error can be approximated through the width of the boxplots and is decreasing to zero with an increasing number of samples. The fact that there are no differences between the error statistics of the weights w_i (compare Fig. 3A–C and D-H) shows that the weights are (i) independent and (ii) all weights are converging to the same analytical PDF. (i) is proven by the fact that all weights show the same pattern of convergence while (ii) is demonstrated through the convergence to zero for large sample sizes.

As a counter example, Fig. 4 compares the proposed sampling strategy (A-C) with a naïve sampling (E-G) where the first weight w_i is uniformly sampled from the unit interval ($w_1 \sim \mathcal{U}[0, 1]$), the second weight is uniformly sampled from the remaining range ($w_2 \sim \mathcal{U}[0, 1 - w_1]$) while the third weight w_3 is set to be the remainder such that the three weights sum up to 1.0 ($w_3 = 1 - w_1 - w_2$). The results show that the naïvely sampled weights do not follow the same distribution. The sampling strategy favors large weights for w_1 and small weights for w_2 and w_3 which leads to a non-uniform sampling of the parameter space. This is depicted in Fig. 4D and H showing the ternary plot of three sampled weights for the proposed method and the naïve method, respectively. It clearly demonstrates that the proposed method samples the domain uniform while the naïve approach preferentially samples large weights for w_1 and low weights for w_2 and w_3 (lower right corner of triangle). The proposed method hence is clearly beneficial when, for example, random samples of the soil texture are drawn by interpreting the weight as the fraction of sand (w_1), silt (w_2) and clay (w_3). The naïve approach samples more sandy soils while only a rare amount of clay and silty soils. The proposed method prevents this.

5. Examples of application of proposed sampling method

The following three sections are to demonstrate examples where the sampling of summative weights is required in real-world environmental modelling applications. These examples are to demonstrate that the proposed sampling is addressing the limitations/pitfalls stated in the introduction. The proposed method is more efficient compared to a state-of-the-practice exclusion-based method (Sec. 5.1), is superior to simple biased sampling approaches (Sec. 5.2), and can be used to avoid sampling artefacts when optimal weights are obtained through automatic calibration (Sec. 5.3).

5.1. Example 1: Efficiency of proposed sampling

A state-of-the-practice method to draw samples of any distribution is the so called accept-reject sampling which samples points within a unit hypercube and rejects samples that do not follow this distribution (Robert and Casella, 2005). In terms of the sampling of summative weights problem this translates into sampling points within a square ($N + 1 = 3$) or cube ($N + 1 = 4$) and withdraw samples that are outside a triangle or tetrahedron, respectively.

Even though this method guarantees that the samples are distributed uniformly within the desired simplex (triangle, tetrahedron), the amount of samples withdrawn increases drastically with increasing dimensions. In case of summative weights, the volume of the unit hypercubes is 1 and the volume of the desired N-simplex is $1/N!$ when $N + 1$ weights are to be sampled (Eq. (2)). This results in only $1/N!$ of the samples drawn within a hypercube being accepted.

Fig. 5A shows that all samples drawn with the proposed method are feasible and uniformly distributed within a ternary plot while Fig. 5B shows that 50% of the samples drawn with the Accept-Reject method are invalid when $N + 1 = 3$ weights are sampled. Fig. 5C shows the non-linear decrease of efficiency (here valid vs. invalid samples). In case $N + 1 = 10$ weights need to be sampled only $1/9! \times 100\% = 0.00028\%$ of the samples drawn are valid. In other words, only around 3 out of 1 000 000 samples are valid. On the contrary, the efficiency of the proposed sampling is 100% independent of the number of weights since the method is constructed to generate samples filling the N-simplex uniformly without rejection.

5.2. Example 2: Sampling of porosity of soil classes

A real-world example of summative weights is to sample soil textures, i.e., sample the sand, silt and clay content of a mineral soil, as for example done by Pozdniakov et al. (2019). These three contents need to sum up to 1. The soil textures are used to define soil classes. Sampled soil textures are often visualized using the soil triangle, which is a ternary plot as shown in Fig. 6A–C and E-G. For demonstration purposes, the exemplary soil class "sandy clay" from the USDA soil triangle, which is of a triangular shape itself, will be sampled from. The soil class "sandy clay" contains soils with sand contents between 45 and 65%, clay contents between 35 and 55% and silt contents up to 20%. Soil textures are often used in hydrologic models and land surface schemes to derive soil properties such as porosity Θ_S in [%] using pedo-transfer functions. For instance, Cosby et al. (1984) obtained the following pedo-transfer function for porosity:

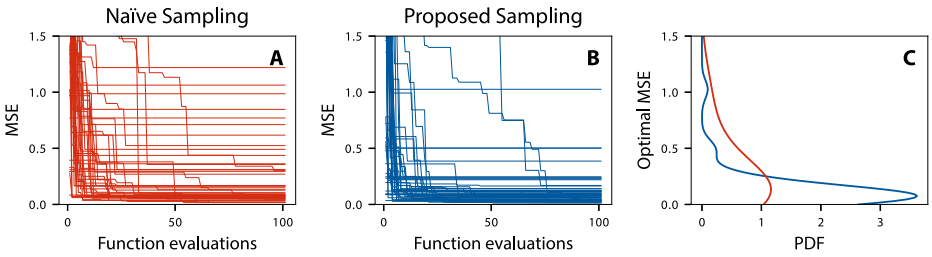


Fig. 7. Calibration of a model that is the weighted average of three options (Eq. (14)). The model hence needs three weights summing up to 1.0 to be sampled. The progression of the mean square error (MSE) used as objective function to minimize is shown over the course of the calibration represented by the number of function evaluations. The budget of all calibration trials using the Dynamically Dimensioned Search algorithm is 100. In total, 50 independent calibration trials were performed using (A) the naïve approach and (B) the proposed sampling to draw the three weights required by the model. (C) The distribution of the final MSE of the 50 trials is based on a kernel density estimate of the 50 sample values.

$$\Theta_S = -0.126 \cdot c_{sand} + 48.9 \tag{13}$$

where c_{sand} represents the percentage of sand in a given soil.

Fig. 6 shows random samples drawn from the soil class "sandy clay" using the proposed (A-C) and the naïve sampling approach (E-F). The difference between the three panels for each approach is solely the "labeling" of the three axis, i.e. the random assignment of which weight corresponds to which soil component. It is expected that this assignment should not matter, which is clearly the case for the proposed sampling method (Fig. 6D). However, in the naïve sampling approach, the corresponding distribution of calculated porosity depends on which weight is assigned to sand, and therefore the leads to a calculated porosity distribution which is (problematically) sensitive to the order of sampling (as shown in Figure 4H).

5.3. Example 3: Calibration of blended model options

So-called blended models are defined as the weighted average of multiple models or model components (Mai et al., 2020). To conserve mass or otherwise respect physical constraints, the weights are required to sum up to 1.

We here use a manufactured example blended model f defined as the weighted average of three components:

$$f(\vec{x}, \vec{p}, \vec{w}) = w_1 \cdot \sin(p_0 \vec{x}) + w_2 \cdot \sin(p_1 \vec{x}) + w_3 \cdot (p_2 + \exp(p_3 \vec{x})) \tag{14}$$

where $\vec{w}$ denotes the weights of the three components, $\vec{p}$ are model component specific parameters, and $\vec{x}$ is an independent variable such as time or space. This model has been contrived to keep the discussion of results straightforward, however these results extend to much more complicated environmental models with summative constraints on their calibrated parameters.

A real-world problem may now be to find the optimal model parameters $\vec{p}$ and weights $\vec{w}$ to fit a set of observations (Chlumsky et al., 2021). Model parameters would be drawn, for example, from a uniform distribution given an upper and lower (physical) limit of the model parameters. The weights need to be sampled such that they sum up to 1 and the sampling distribution should be identical so as to not favour one model component over another one.

Fig. 7A shows the results of such a calibration in case a naïve sampling is used to find the optimal weights (and parameters) while Fig. 7B shows the same experiment but using the proposed sampling for the weights. One vector of synthetic "observations" at 5000 points, $f'(\vec{x})$, was derived using a fixed set of parameters and weights for all experiments and trials. The optimum of the objective function is hence known to be 0 in this synthetic experiment. In total 50 independent trials have been performed for each of the two experiments. The fixed computational budget Dynamically Dimensioned Search (DDS) (Tolson and Shoemaker, 2007) algorithm was used for automatic calibration. A budget of 100 function evaluations was used in each trial.

The distribution of the best objective function value based on the 50 trials (Fig. 7C) demonstrates that the proposed sampling is more likely to find better objective function values after 100 iterations. The sampling approach described here will be similarly useful for other optimization algorithms based upon stochastic search, such as simulated annealing (Kirkpatrick et al., 1983) or shuffled complex evolution (SCE) (Duan et al., 1992) algorithms, but not for deterministic gradient-based optimization methods as they are not based on random sampling.

6. Conclusions

A simple and effective strategy for sampling $N + 1$ summative weights from identical distributions is derived and demonstrated to ensure appropriate methods are known to environmental modellers. The method is shown to be effective and perfectly unbiased for multiple values of N. Three example applications demonstrate the computational efficiency and superiority of this method compared to other sampling approaches. The approach may be used any time where uniform random sampling of variables with a summation constraint is needed; such problems are found in a wide variety of calibration, uncertainty analysis, and sensitivity analysis exercises. Python and R implementations of the sampling algorithm of equation (12) are freely available on GitHub under https://github.com/julemai/PieShareDistribution Mai et al. (2020).

We recommend readers that require their weights to sum up to values other than 1 or requiring additional lower and upper bounds for the weights, to look into the sampling method introduced by Griffin et al. (2020) as it addresses these additional constraints. Besides these limitations, i.e., fixed upper/lower bounds and the sum being 1, the authors are not aware of any other limitation or challenge associated with the use of the sampling method proposed.

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.

Acknowledgments

This research was undertaken thanks primarily to funding from the Canada First Research Excellence Fund provided to the Global Water Futures (GWF) Project and the Integrated Modeling Program for Canada (IMPC).

References

Arsenault, R., Gatien, P., Renaud, B., Brissette, F., Martel, J.-L., 2015. A comparative analysis of 9 multi-model averaging approaches in hydrological continuous streamflow simulation. J. Hydrol. 529, 754–767.

Bini, E., Buttazzo, G.C., 2005. Measuring the performance of schedulability tests. R. Time Syst. 30, 129–154.

Chlumsky, R., Mai, J., Craig, J.R., Tolson, B.A., 2021. Simultaneous calibration of hydrologic model structure and parameters using a blended model. Water Resour. Res. 57, e2020WR029229.

Cosby, B.J., Hornberger, G.M., Clapp, R.B., Ginn, T.R., 1984. A statistical exploration of the relationships of soil moisture characteristics to the physical properties of soils. Water Resour. Res. 29, 682–690.

Duan, Q., Sorooshian, S., Gupta, V., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour. Res. 28, 1015–1031.
Ganji, A., Maier, H.R., Dandy, G.C., 2016. A modified Sobol' sensitivity analysis method for decision-making in environmental problems. Environ. Model. Software 75, 15–27.
Griffin, D., Bate, I., Davis, R.I., 2020. Generating utilization vectors for the systematic evaluation of schedulability tests. In: IEEE Real-Time Systems Symposium. University of York, pp. 1–14.
Hyde, K.M., Maier, H.R., 2006. Distance-based and stochastic uncertainty analysis for multi-criteria decision analysis in Excel using Visual Basic for Applications. Environ. Model. Software 21, 1695–1710.
Hyde, K.M., Maier, H.R., Colby, C.B., 2004. Reliability-based approach to multicriteria decision analysis for water resources. J. Water Resour. Plann. Manag. 130, 429–438.
Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.
Mai, Juliane, Craig, James R., Tolson, Bryan A., 2020. The Pie sharing problem: Unbiased sampling of N+1 summative weights. (v1.0). Zenodo. https://doi.org/10.5281/zenodo.4300332.
Mai, J., Craig, J.R., Tolson, B.A., 2020. Simultaneously determining global sensitivities of model parameters and model structure. Hydrol. Earth Syst. Sci. 24, 5835–5858.
Moeini, A., Abbasi, B., Mahlooji, H., 2011. Conditional distribution inverse method in generating uniform random vectors over a simplex. Commun. Stat. Simulat. Comput. 40, 685–693.
Montanari, A., Di Baldassarre, G., 2013. Data errors and hydrological modelling: the role of model structure to propagate observation uncertainty. Adv. Water Resour. 51, 498–504.
Piotrowski, A.P., Osuch, M., Napiorkowski, J.J., 2019. Joint optimization of conceptual rainfall-runoff model parameters and weights attributed to meteorological stations. Water Resour. Manag. 33, 4509–4524.
Pozdniakov, S.P., Wang, P., Lekhov, V.A., 2019. An approximate model for predicting the specific yield under periodic water table oscillations. Water Resour. Res. 55, 6185–6197.
Robert, C.P., Casella, G., 2005. Monte Carlo Statistical Methods, second ed. Springer texts in statistics, Springer, Berlin. URL: https://cds.cern.ch/record/1187871.
Stafford, R., 2006. Random vectors with fixed sum. URL: https://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum.
Tolson, B.A., Shoemaker, C.A., 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. Water Resour. Res. 43, W01413.