On the number of Monte Carlo runs in comparative probabilistic LCA

Heijungs, R.

The International Journal of Life Cycle Assessment
https://doi.org/10.1007/s11367-019-01698-4

UNCERTAINTIES IN LCA

On the number of Monte Carlo runs in comparative probabilistic LCA

Reinout Heijungs [1,2]

Received: 14 May 2019 / Accepted: 8 October 2019

Abstract

Introduction The Monte Carlo technique is widely used and recommended for including uncertainties LCA. Typically, 1000 or 10,000 runs are done, but a clear argument for that number is not available, and with the growing size of LCA databases, an excessively high number of runs may be a time-consuming thing. We therefore investigate if a large number of runs are useful, or if it might be unnecessary or even harmful.

Probability theory We review the standard theory or probability distributions for describing stochastic variables, including the combination of different stochastic variables into a calculation. We also review the standard theory of inferential statistics for estimating a probability distribution, given a sample of values. For estimating the distribution of a function of probability distributions, two major techniques are available, analytical, applying probability theory and numerical, using Monte Carlo simulation. Because the analytical technique is often unavailable, the obvious way-out is Monte Carlo. However, we demonstrate and illustrate that it leads to overly precise conclusions on the values of estimated parameters, and to incorrect hypothesis tests.

Numerical illustration We demonstrate the effect for two simple cases: one system in a stand-alone analysis and a comparative analysis of two alternative systems. Both cases illustrate that statistical hypotheses that should not be rejected in fact are rejected in a highly convincing way, thus pointing out a fundamental flaw.

Discussion and conclusions Apart form the obvious recommendation to use larger samples for estimating input distributions, we suggest to restrict the number of Monte Carlo runs to a number not greater than the sample sizes used for the input parameters. As a final note, when the input parameters are not estimated using samples, but through a procedure, such as the popular pedigree approach, the Monte Carlo approach should not be used at all.

Keywords Accuracy · Life cycle assessment · Monte Carlo · Precision · Uncertainty

1 Introduction

Uncertainty in LCA is pervasive, and it is widely acknowledged that uncertainty analyses should be carried out in LCA to grant a more rigorous status to the conclusions of a study (ISO 2006, JRC-IES 2010). The most popular approach for doing an uncertainty analysis in LCA is the Monte Carlo approach (Lloyd and Ries 2007), partly because it has been implemented in many of the major software programs for LCA, typically as the only way for carrying out uncertainty analysis (for instance, in SimaPro, GaBi, Brightway2, and in openLCA).

The Monte Carlo method is a sampling-based method, in which the calculation is repeated a number of times, in order to estimate the probability distribution of the result (see, e.g., Helton et al. 2006, Burmaster and Anderson 1994). This distribution is then typically used to inform decision-makers about characteristics, such as the mean value, the standard deviation or quantiles (such as the 2.5 and 97.5 percentiles). In LCA, the results are typically inventory results (e.g., emissions of pollutants) or characterization/normalization results (e.g., climate change, human health, etc.). In comparative LCA, such distributions form the basis of paired comparisons and tests of hypothesis (Mendoza Beltran et al. 2018). Many programs and studies offer or present visual aids for interpreting the results, including histograms and boxplots (Helton et al. 2006; McCleese and LaPuma 2002).

Responsible editor: Yi Yang

✉ Reinout Heijungs
r.heijungs@vu.nl

[1] Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands

[2] Institute of Environmental Sciences (CML), Leiden University, Einsteinweg 2, 2333 CC Leiden, The Netherlands

Published online: 22 October 2019

A disadvantage of the Monte Carlo method is that it can be computationally expensive. Present-day LCA studies can easily include 10,000 or more unit process, and calculating such as system can take some time. Repeating this calculating for a new configuration then takes the same time, and this is repeated a large number of times. Finally, the stored results must be analyzed in terms of means, standard deviations, p values and visual representations. Altogether, if we use the symbol N_{run} to refer to the number of Monte Carlo runs, the symbol T_{cal} for the CPU time needed to do one LCA calculation, and T_{ana} for the time needed to process the Monte Carlo results, the total time needed, T_{tot}, is simply

$$T_{\text{tot}} = N_{\text{run}} \times T_{\text{cal}} + T_{\text{ana}}$$

Usually, $T_{\text{cal}} > T_{\text{ana}}$ and certainly $N_{\text{run}} \times T_{\text{cal}} \gg T_{\text{ana}}$, so that we can write

$$T_{\text{tot}} \approx N_{\text{run}} \times T_{\text{cal}}$$

and further ignore the aspect of T_{ana}.

The time needed for a Monte Carlo analysis is thus determined by two factors: T_{cal}, which is typically in the order of seconds or minutes, and N_{run}. A normal practitioner has little influence on T_{cal}, as it is dictated by the combination of algorithm, the hardware, and the size of the database. Typically, it is between 1 s and 5 min. (This is a personal guess; there is no literature on comparative timings using a standardized LCA system). A practitioner has much more influence on the number of Monte Carlo runs, N_{run}. So, the trick is often to take N_{run} not excessively high, say 100 or 1000. On the other hand, it has been claimed that this number must be large, for instance 10,000 or even 100,000. For instance, Burmaster and Anderson (1994) suggest that "the analyst should run enough iterations (commonly ≥10,000)," and the authoritative Guide to the Expression of Uncertainty in Measurement (BIPM 2008) writes that "a value of $M = 10^6$ can often be expected to deliver [a result that] is correct to one or two significant decimal digits." In the LCA literature, we find similar statements, for instance by Hongxiang and Wei (2013) ("more than 2000 simulations should be performed") and by Xin (2006) ("[it] should run at least 10,000 times"). Such claims also end up in reviewer comments: We recently received the comment "Monte Carlo experiments are normally run 5000 or 10,000 times. In the paper, Monte Carlo experiments are only run 1000 times. Explain why?". With the pessimistic T_{cal} = 5 min, using N_{run} = 100,000 runs will require almost 1 year. If we take the short calculation time of T_{cal} = 1 s, we still need more than one full day. And, even Brightway2's (https://brightwaylca.org/) claim of "more than 100 Monte Carlo iterations/second" (of which we do not know if this also applies to today's huge systems) would take more than 16 min. Such waiting times may be acceptable for Big Science, investigating fundamental questions on the Higgs boson or the human genome. But, for a day-by-day LCA consultancy firm, even 1 h is much too long.

In this study, we investigate the role of N_{run}. We will in particular focus on the original purpose of the Monte Carlo technique vis-à-vis its use in LCA, and consider the fact that in LCA, the input probability distributions are often based on small samples, or on pedigree-style rules-of-thumb, as well as the fact that in LCA, we are in most cases interested in making comparative statements ("product A is significantly better than product B").

The next section discusses the elements of the analysis: the mathematical model and its probabilistic form, the description of probabilistic ("uncertain") data, the estimation of input data, and the estimation of output results. Section 3 provides two numerical examples. Section 4 finally discusses and concludes.

2 Probability theory

In this section, we discuss a few background topics from probability theory. The interested reader is referred to general textbooks, such as Ross (2010) and Gharamani (2005).

2.1 Mathematical models

When a model needs several input variables to compute an output variable, we can abstractly write the model relation as

$$y = f(x_1, x_2, \ldots)$$

Here, x_1, x_2, … represent the values of the input variables (the data, for instance CO_2 coefficients and electricity requirements) and y is the output (the result, for instance a carbon footprint). The function $f(\cdot)$ is a specification of the LCA algorithm (Heijungs and Suh 2002). We will assume that this algorithm is known and fixed, and that it has been implemented in software in a reliable way and therefore does not introduce any uncertainty (however, see Heijungs et al. 2015).

2.2 Probabilistic models

Uncertainty can enter the scene in different ways:

- When the input data is not exactly known (for instance, the effect of glyphosate on human health is not fully known)
- When the input data displays variability (for instance, the lifetime of identical light bulbs is not exactly equal)
- When choices must be made by the analyst (for instance, allocation factors can be based on mass or on economic value)

Sometimes, additional sources of uncertainty are mentioned (Huijbregts 1998), such as model uncertainty. Here, we restrict the discussion to those types of uncertainty that can be phrased as inputs (x_1, x_2, etc.) in the model equation ($f(\cdot)$). Our analysis can, however, easily be broadened to cover such cases. For instance, we can include allocation choices as an extra input parameter into $f(\cdot)$. (Heijungs et al. 2019).

2.3 Probability distributions of input variables

In a probabilistic model, we can specify the input data as a probability distribution (continuous or discrete). So, from now on, we will assume that x_1, x_2, … are not fixed numbers, but that they are stochastic (random) numbers, following some probability distribution. We will use the convention from probability theory to indicate stochastic variables with capital letters, like X_1, X_2, … Further, the symbol ~ indicates that a stochastic variable is distributed according to some probability distribution. For instance,

$$\begin{cases} X_1 \sim N\left(\mu_{X_1}, \sigma_{X_1}\right) \\ X_2 \sim N\left(\mu_{X_2}, \sigma_{X_2}\right) \\ \cdots \sim \cdots \end{cases}$$

where $N(\mu, \sigma)$ is the normal (Gaussian) probability distribution with parameters μ and σ. We might go for other probability distributions (uniform, log-normal, binomial, etc.) but at this stage want to keep the discussion simple. The numbers that specify the numerical details of the probability distribution (here μ and σ in general, and more specifically μ_{X_1}, μ_{X_2}, σ_{X_1}, σ_{X_2}, etc.) are referred to as parameters. So, not x_1 is a parameter (as the usual terminology in LCA goes), but rather μ_{X_1} and σ_{X_1} are parameters of the distribution of X_1. Other types of distributions are usually specified with different types of parameters (for instance, the uniform distribution with a parameter for the lower limit and a parameter the upper limit) or even with another number of parameters (for instance, the Poisson distribution requires only one parameter, while the asymmetric triangular distribution requires three parameters).

2.4 Probability distributions of output variables

Recognizing that (some of) the input parameters of the model $f(\cdot)$ are stochastic, a logical consequence is that the model output is also stochastic. Thus, we write

$$Y = f(X_1, X_2, \ldots)$$

See Heijungs et al. (2019). With this change of y into Y, our task shifts from calculating the value of y to calculating the distribution of Y. More specifically, we may want to know:

- The shape of the distribution of Y (i.e., normal, uniform, log-normal, binomial, etc.)
- The value or values of the parameter or parameters (e.g., μ_Y and σ_Y)

Probability theory offers methods to calculate the probability distribution of Y when those of X_1, X_2, … are given, but only for a few cases of $f(\cdot)$ and only for a few input distributions. For instance, when $Y = f(X_1, X_2) = X_1 + X_2$ and X_1 and X_2 are normal, every textbook shows that

$$Y \sim N\left(\mu_{X_1} + \mu_{X_2}, \sqrt{\sigma_{X_1}^2 + \sigma_{X_2}^2}\right)$$

In words, the sum of two normal variables is itself normally distributed, and the parameters μ_Y and σ_Y can easily be calculated from the parameters of the input distributions. Another case is $Y = f(X_1, X_2) = X_1^2 + X_2^2$. This is pretty complicated, but when we take the special case of $\mu_{X_1} = \mu_{X_2} = 0$ and $\sigma_{X_1} = \sigma_{X_2} = 1$, it is a well-known result:

$$Y \sim \chi^2(2)$$

where $\chi^2(\nu)$ is the chi-squared distribution with parameter ν. In general, most choices of $f(\cdot)$ with less trivial combinations of X_1, X_2, … (such as $f(X_1, X_2) = X_1 X_2^2 + \frac{\ln X_1}{4 + \sin X_2}$) are not manageable by the theory of probability. It is therefore important to have an alternative way to determine the probability function of such more complicated functions of stochastic variables. The same applies also to situations where $f(\cdot)$ is straightforward, but where the input distributions for X_1, X_2, etc. are not normal.

The Monte Carlo approach (Metropolis and Ulam 1949; Shonkwiler and Mendivil 2009) can be used as an alternative way for constructing the probability distribution of Y in case the mathematical approach is too hard. It is based on artificially sampling values from Y, and using this sample for reconstructing (the technical term is estimating) the shape and the parameter values of Y. We will spend the next section on the topic of estimating a probability distribution from a sample of values. This is a topic of more general interest than Monte Carlo simulations, so we will keep the discussion quite general, also covering the case of estimating the distribution of input variables like X_1 and X_2.

2.5 Estimating a probability distribution in general

We will discuss the question of estimating a probability distribution Z (including its parameters), given a sample of data, z_1, z_2, …, z_n. This task is known as the estimation problem, and it is one of the central topics of inferential statistics. See, for instance, Rice (2007) and Casella and Berger (2002) for general textbooks.

Suppose we have a sample of data from an unknown stochastic process, Z. Let the sampled values be indicated by z_i, for $i = 1, \ldots, n$. If we want to estimate the probability

distribution belonging to the stochastic process that generated this sample, we must first make an assumption about the type of distribution. Is it a normal distribution, a uniform distribution, a log-normal distribution, a Weibull distribution? This choice is one of the trickiest parts of the entire estimation process, because there is no clear guidance. Different aspects can play a role here:

- Evidence: the data (e.g., a histogram or a boxplot) may suggest a certain distribution.
- Conventions and compatibility with software: the log-normal distribution has a longer and more widespread history in LCA than the Erlang distribution.
- Familiarity and simplicity: if the histogram looks approximately bell-shaped, a normal distribution is more natural than the Cauchy distribution.
- Statistical criteria: we can use statistical tests (such as those by Kolmogorov-Smirnov and Anderson-Darling) to assess the goodness-of-fit with a number of probability distributions.

Clearly, there are also cases where none of the conventional model distributions provides a satisfactory fit with the empirical data. We will not further discuss such cases, because the usual procedure in LCA is to model input uncertainties in terms of just a few distributions: lognormal, normal, uniform, or triangular (Frischknecht et al. 2004) or perhaps a few more (gamma and beta PERT; see Muller et al. 2016).

Once we have selected a probability distribution, the next task is to estimate the parameter value or values of that distribution. Suppose we have selected a normal distribution, so

$$Z \sim N(\mu_Z, \sigma_Z)$$

where μ_Z and σ_Z are the distribution's parameters, which are still unknown at this stage of the analysis. Then, our task is to estimate the values of μ_Z and σ_Z that correspond best with the sampled data. Different estimation principles are available in the statistical literature to do this. Two widely used principles are the method of moments and the method of maximum likelihood. For the case of a normal distribution, these two principles yield the same estimate of μ_Z and σ_Z, but for some distributions, there is a difference in the outcome of the estimation procedure. Anyhow, the theory of statistics offers formulas for estimators, which are functions of the observations. We can use the symbol of the parameter to be estimated with a hat on top of it to indicate the estimator: $\hat{\mu}$ is an estimator of μ and $\hat{\sigma}$ is an estimator of σ. In the case of a normal distributions, both estimation principles (method of moments and method of maximum likelihood) suggest

$$\hat{\mu}_Z = \frac{1}{n}\sum_{i=1}^{n} Z_i$$

and

$$\hat{\sigma}_Z = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Z_i - \hat{\mu}_Z\right)^2}$$

as estimators for μ_Z and σ_Z. When applied to a concrete data set, $z_1, z_2, \ldots, z_n$, these estimators produce a concrete value, because we insert the observed values of z_i at the place of the stochastic variable Z_i. These concrete values are the estimates, which we will indicate hereafter as $\bar{z}$ and s_Z.

Of course, we cannot expect that the estimates will be fully accurate if the sample size is finite. The estimate $\bar{z}$ will be hopefully close to the true value μ_Z, but probably it will be a little bit off (that is also why we distinguish the symbols: in general $\bar{z} \neq \mu_Z$, but $\bar{z} \approx \mu_Z$). The same applies to the estimate s_Z of σ_Z.

The theory of inferential statistics not only allows to estimate the values, but it also allows us to say something about the level of precision of such estimates. This is done through the theory of sampling distributions, standard errors, and confidence intervals.

A sampling distribution is the probability distribution of an estimator. Let us suppose we have a probability distribution $Z \sim N(\mu_Z, \sigma_Z)$, with unknown parameter μ_Z and known parameter σ_Z, from which we sample n observations, and use the estimator $\hat{\mu}_Z$ to estimate μ_Z by the value $\bar{z}$. If we would take another sample of size n, we can use the same estimator to again estimate μ_Z, but we will find a slightly different value $\bar{z}$, because the sample will contain different values. Repeating and repeating, always with the same sample size n, we will end up with a distribution of $\bar{z}$ values. This distribution will be referred to as $\overline{Z}$.

The famous central limit theorem states that the distribution of the estimates of the mean, $\overline{Z}$, is normally distributed and that there is a simple relation between its parameters ($\mu_{\overline{Z}}$ and $\sigma_{\overline{Z}}$) and the parameters of the parent distribution Z (μ_Z and σ_Z):

$$\overline{Z} \sim N\left(\mu_Z, \frac{\sigma_Z}{\sqrt{n}}\right)$$

So, $\mu_{\overline{Z}} = \mu_Z$ and $\sigma_{\overline{Z}} = \frac{\sigma_Z}{\sqrt{n}}$. This first fact signifies that the mean of the sample means corresponds to the mean of the parent distributions. This is a convenient property, because it allows to use the sample mean ($\bar{z}$) as the best guess of μ_Z. The second fact tells us that the width of the distribution of $\overline{Z}$ (so $\sigma_{\overline{Z}}$) depends on the width of the distribution of Z (so on σ_Z) and on the size of the sample (so on n). In fact, $\sigma_{\overline{Z}}$ decreases without limits when n increases. The important consequence is that the estimate of μ_Z, $\bar{z}$, is more precise when n is large and that we can determine its value as precisely as we want by just increasing sample size. The larger the sample, the more precise the estimate.

The quantity $\sigma_{\overline{Z}} = \frac{\sigma_Z}{\sqrt{n}}$ is known as the standard error of the mean, also known as "the" standard error. For a precise estimation of μ_Z, we want this $\sigma_{\overline{Z}}$ to be small. The only way to do so is to use a large sample size n, because σ_Z is fixed. The standard error is related to the concept of a confidence interval. For the case of estimating μ_Z, the 95% confidence interval is given by

$$CI_{\mu_Z;0.95} = \left[\overline{z}-1.96\sigma_{\overline{Z}}, \overline{z}+1.96\sigma_{\overline{Z}}\right]$$

This means that with 95% confidence, the interval *CI* will contain the true value μ_Z that we are supposed to estimate by $\overline{z}$. Observe that the confidence interval has a width of $2 \times 1.96\sigma_{\overline{Z}} = 3.92\sigma_{\overline{Z}} = 3.92\frac{\sigma_Z}{\sqrt{n}}$. If we want this interval to be smaller, we need to increase sample size n.

Above, we discussed how to estimate the parameter μ when the parameter σ is known. Estimation of σ and other parameters, and estimation of μ when σ is unlnown, are technically more difficult, but conceptually the idea is the same.

2.6 Estimating the probability distribution of input variables

When we want to estimate the probability distribution of an input variable (X_1, etc.), we carry out the following steps:

- We sample data ($x_{11}, x_{12}, \ldots, x_{1n}$) from the phenomenon (e.g., unit process).
- We choose a convenient probability distribution shape (e.g., normal).
- We use the formulas for the estimators ($\hat{\mu}_{X_1}$, $\hat{\sigma}_{X_1}$, etc.) to find estimates ($\overline{x_1}$, s_{X_1}, etc.).

The estimated parameter values ($\overline{x_1}$, s_{X_1}, etc.) are "best guesses" given the available data. However, we cannot expect that they are perfect estimates, because the confidence interval of these parameters decreases with $\frac{1}{\sqrt{n}}$, and n is usually limited. Of course, we can increase n by collecting more primary data, but site visits and measurements are usually expensive and time-consuming. For that reason, in LCA, as in most other fields of science, n is usually quite limited. The price we pay for that is a larger standard error and a wider confidence interval.

2.7 Estimating the probability distribution of output variables, given perfectly known inputs

Next, we move to the topic of estimating the probability distribution of an output variable (Y, etc.). Suppose, for simplicity, we have one stochastic input variable, X, normally distributed, with known parameters:

$$X \sim N(\mu_X, \sigma_X)$$

Next, we define a very simple function of that variable:

$$Y = f(X) = X$$

Of course, the distribution of the output variable Y is trivial:

$$Y \sim N(\mu_X, \sigma_X)$$

and in particular, $\mu_Y = \mu_X$. But, let us pretend we are bad in probability theory and prefer to use a Monte Carlo approach. We simulate N_{run} instances of X (namely $x_1, x_2, \ldots, x_{N_{\text{run}}}$) and use that to calculate N_{run} instances of Y (namely $y_1 = x_1$, $y_2 = x_2$, etc.). These values of y are used to estimate μ_Y as follows:

$$\overline{y} = \frac{1}{N_{\text{run}}} \sum_{i=1}^{N_{\text{run}}} y_i$$

When the sample has been obtained in a random way, we can also be sure that the estimate will converge to the correct value:

$$\lim_{N_{\text{run}} \to \infty} \overline{y} = \mu_Y = \mu_X$$

Likewise, we can estimate the standard deviation of Y, σ_Y. This can be used to find the standard error of the mean

$$s_{\overline{Y}} = \frac{s_Y}{\sqrt{N_{\text{run}}}}$$

The noteworthy aspect of this standard error is that it will go to zero when N_{run} grows very large:

$$\lim_{N_{\text{run}} \to \infty} s_{\overline{Y}} = 0$$

As a consequence, the estimate of μ_Y will become arbitrarily precise, if we have enough computer time:

$$\lim_{N_{\text{run}} \to \infty} CI_{\mu_Y;0.95} = [\mu_Y, \mu_Y] = [\mu_X, \mu_X]$$

That is not surprising. If we would have been more thoughtful, we could have saved the computer expenses and directly deduce that $\mu_Y = \mu_X$, with infinite precision. The situation is comparable to computing $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \ldots$, for a large number of terms, or being more thoughtful and directly writing this as $\frac{\frac{1}{2}}{1-\frac{1}{2}} = 1$. Both approaches yield approximately the same result. So, when we want to use a Monte Carlo approach to estimate the parameters of a probability distribution, we must use a large sample size N_{run} to find a reliable answer. The recommendations quoted in the introduction (1000, 10,000, 100,000) are based on the situation described

here: accurately estimating an output distribution on the basis of perfect knowledge of the input distributions.

2.8 Estimating the probability distribution of output variables, given imperfectly known inputs

But now, take the next case, a normal distribution with parameters μ_X and σ_X, but under the provision that μ_X itself is slightly off, because we did not know μ_X but used its imperfect estimate $\bar{x}$. So, we consider

$$X \sim N\left(\bar{x}, \sigma_X\right)$$

Next, we again study the trivial function

$$Y = f(X) = X$$

first analytically, using probability theory, and then through a Monte Carlo simulation.

Analytically, we find

$$Y \sim N\left(\bar{x}, \sigma_X\right)$$

The essential point to observe is that the mean of Y is not μ_X but $\bar{x}$, which is likely to be somewhat wrong.

Next, let us try this by a Monte Carlo simulation. We use $\bar{y}$ to estimate μ_Y. It will be close to $\bar{x}$, rather than close to μ_X. Moreover, the standard error of this estimate is still $s_{\bar{Y}} = \frac{s_Y}{\sqrt{N_{run}}}$, so as close to 0 as we like. In fact,

$$\lim_{N_{run} \to \infty} CI_{\mu_Y;0.95} = [x, x]$$

Summarizing, using probability theory and using the Monte Carlo approach, both will give you the wrong value ($\bar{x}$ instead of μ_X) when estimating μ_Y, and the Monte Carlo approach will in addition suggest that this estimate is very precise due to a vanishing standard error, at least when N_{run} is very large.

Observe that this is not a mistake or limitation of the Monte Carlo approach. In fact, it performs very well. The mistake is entirely due to the analyst, who uses an imperfectly estimated input parameter ($\bar{x}$ instead of μ_X) to run an infinite-precision method. Also, observe that this is a very ubiquitous situation in LCA: Most LCA data on unit processes is obtained from limited samples. Even a sample size of 1 is not uncommon. There is even a widely used approach, referred to as the pedigree approach and popularized by the ecoinvent database, of which the purpose is to estimate a probability distribution on limited data (Frischknecht et al. 2004; Weidema et al. 2013). We devote a longer discussion to this problem toward the end of this paper.

3 Numerical illustration

To test and illustrate these ideas, we did two simulation experiments, first for one stand-alone system, and then for two systems in a comparative analysis.

To illustrate the situation for one system, we made a small code in R (Fig. 1) and used it to simulate the following case:

- The parent distribution is $X \sim N(10, 1)$.
- We sample $n = 16$ observations, and estimate μ_X by $\bar{x}$.
- We draw from $Y \sim N(\bar{x}, \sigma_X)$ a Monte Carlo sample of size $N_{run} = 100{,}000$.
- From this sample, we estimate μ_Y by $\bar{y}$.

In our simulation, the results were as follows:

- $\bar{x} = 10.31$, $\sigma_{\bar{X}} = 0.25$, so the 95% confidence interval for μ_X is [9.819, 10.799].
- $\bar{y} = 10.31$, $\sigma_{\bar{Y}} = 0.0031$, so the 95% confidence interval for μ_Y is [10.305, 10.318].

The interpretation of these results are as follows:

- We misestimate μ_X (10.31 instead of 10.00).
- But, we acknowledge that it may be wrong, and in fact, our 95% confidence interval contains the correct value (it suggests a value somewhere between 9.8 and 10.8).
- We misestimate μ_Y (10.31 instead of 10.00).
- But, we deny that it may be wrong, because our 95% confidence interval is pretty sure about a value somewhere 10.30 and 10.32.

In conclusion, the Monte Carlo approach will yield a very precise, but inaccurate, result.

Fig. 1 R code for generating a large Monte Carlo sample (N_{run} = 100,000) from an input distribution with limited precision

```
mu_x <- 10 # define population mean for X
sigma_x <- 1 # define population standard deviation for X
n <- 16 # define sample size for X
x <- rnorm(n, mu_x, sigma_x) # generate vector of random numbers for X
x_bar <- mean(x) # estimate mean of X
N_run <- 100000 # define number of Monte Carlo runs
y_mc <- rnorm(N_run, x_bar, sigma_x) # generate the Monte Carlo result
# compute 95% confidence interval
c(mean(y_mc)-1.96*sd(y_mc)/sqrt(N_run), mean(y_mc)+1.96*sd(y_mc)/sqrt(N_run))
```

The precision of an estimate plays an important role in testing statistical hypotheses. When we would like to test a statement like $\mu_X = 10$, the null hypothesis significance testing procedure would not reject the null hypothesis, because the hypothesized value of 10 is in the 95% confidence interval [9.819, 10.799]. On the other hand, the same procedure when applied to the null hypothesis $\mu_Y = 10$ would lead to a rejection, because 10 is not in the 95% confidence interval [10.305, 10.318].

The second example is about two systems, A and B, in a comparative LCA: Seemingly precise estimates of the impact of products A and B can lead to the conclusion that A is better than B, while the real situation is that B is better than A. Or we find that A is better than B, although they do not differ. To test and illustrate this phenomenon, we made another computer experiment (Fig. 2). We generate $n = 16$ samples from $X_A \sim N(10, 1)$ and $X_B \sim N(10, 1)$. From these two samples, we estimate μ_{X_A} through $\bar{x}_A$ and μ_{X_B} through $\bar{x}_B$ and do a two-sample t test to test the hypothesis $\mu_{X_A} = \mu_{X_B}$. Next, we use $Y_1 = f(X_1) = X_1$ and $Y_2 = f(X_2) = X_2$, and sample $N_{run} = 100{,}000$ values from Y_A and Y_B. From this Monte Carlo sample, we test the null hypothesis $\mu_{Y_A} = \mu_{Y_B}$. The p value of the first test was 0.67 providing strong evidence of equality of μ_{X_A} and μ_{X_B}. The second test yielded a p value around 10^{-16}, pointing to overwhelming evidence that $\mu_{Y_A} \neq \mu_{Y_B}$.

This comparative case is even more interesting than the first example, because decisions about purchases, ecolabels, etc. are often taken on the basis of comparative assessments: Is there evidence that one product is significantly better than another product? Statistical hypothesis testing can provide an answer to such questions, but the example shows that inaccurately specified parameters of the parent distributions may give a seemingly convincing wrong answer, because an excessive number of Monte Carlo runs will optimize precision, ignoring inaccurate inputs.

4 Discussion and conclusions

Let us be a bit more explicit on the terminology: An estimate can be imprecise or it can be inaccurate. The two have been illustrated in various ways (Fig. 3). In our analysis of example 1, we have an inaccurate estimate ($\bar{y}$ can be off quite a bit due to small n in determining $\bar{x}$) with arbitrary high precision ($\sigma_{\bar{Y}}$ is almost zero due to very large N_{run}). By reporting a very small standard error of the mean, we suggest to have done a high-quality calculation.

The discussion above took a very trivial function, namely $Y = f(X) = X$ as starting point. The storyline is no different for more complicated cases, such as $Y = f(X_1, X_2) = X_1 X_2^2 + \frac{\ln X_1}{4+\sin X_2}$ or for functions of hundreds of input distributions $Y = f(X_1, X_2, \ldots)$. Likewise, we used a normal distribution with known standard deviation to start with. If the standard deviation is unknown, or if the parent distribution is of a different type (log-normal, binomial, ...), the mathematics is more difficult, but the take home message remains the same: with an imprecise estimate of the input parameters, we can make a very precise but probably inaccurate estimate of the output parameters. Garbage in, garbage out, but the type of garbage has changed: from imprecise to inaccurate. That is a problem, because imprecision is visible through a large standard error of the mean ($\bar{x} = 10.31 \pm 0.25$), while inaccuracy is not visible ($\bar{y} = 10.31 \pm 0.0031$). As a result, the estimate will suggest to be of high quality where it is not.

Superficially, it sounds better to make precise statements than imprecise statements. But, when the statements are on inaccurate values, this is not necessarily true.

In a statistical analysis, we can always draw wrong conclusions (type I errors: not rejecting an incorrect null hypothesis, type II errors: rejecting a correct null hypothesis), but this is a completely different type of error: rejecting a null hypothesis for which we have no appropriate data. The root of the problem is that we sample from inaccurately specified distributions. While we would naively expect that this leads to inaccurate results, the statistical analysis neglects the inaccuracy and concentrates on the precision. The imprecision declines with the number of Monte Carlo runs, but the inaccuracy does not. And, imprecision is visible, while inaccuracy is invisible.

The remedy is to maintain the imprecision in the estimate of the input parameters. As long as the parameters of the input distributions are imprecise, we should not be allowed to decrease the precision of the output distribution estimates

```
mu_xA <- 10 # define population mean for XA
sigma_xA <- 1 # define population standard deviation for XA
n_xA <- 16 # define sample size for XA
xA <- rnorm(n_xA, mu_xA, sigma_xA) # generate vector of random numbers for XA
mu_xB <- 10 # define population mean for XB
sigma_xB <- 1 # define population standard deviation for XB
n_xB <- 16 # define sample size for XB
xB <- rnorm(n_xB, mu_xB, sigma_xB) # generate vector of random numbers for XB
xA_bar <- mean(xA) # estimate mean of XA
xB_bar <- mean(xB) # estimate mean of XB
t.test(xA, xB) # testing equality of means of XA and XB
N_run <- 100000 # define number of Monte Carlo runs
yA <- rnorm(N_run, xA_bar, sigma_xA) # generate the Monte Carlo result for YA
yB <- rnorm(N_run, xB_bar, sigma_xB) # generate the Monte Carlo result for YB
t.test(yA, yB) # testing equality of means of YA and YB
```

Fig. 2 R code for testing the hypothesis of equality of means in the input data X_1 and X_2, generated from small samples ($n_{X_A} = n_{X_B} = 16$), and of equality of means in the output results Y_1 and Y_2, generated with a large Monte Carlo sample (N_{run} = 100,000)

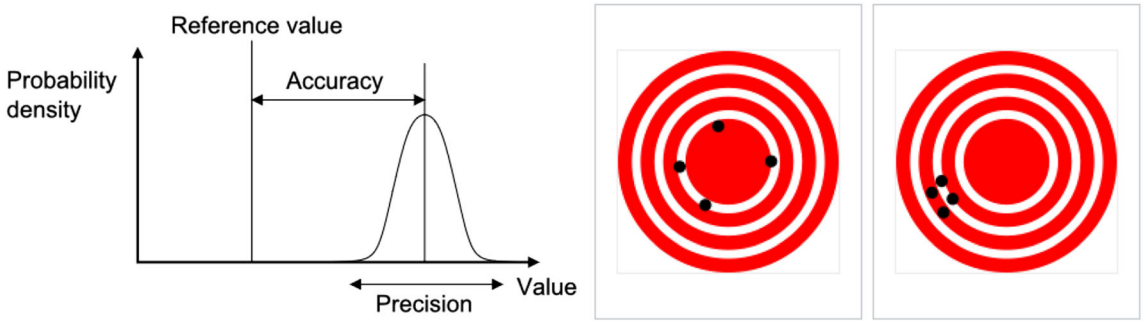

Fig. 3 Illustration of the difference between precision and accuracy. The left figure illustrates both; the middle one is an example of low precision and the right one is an example of low accuracy. Source: https://en.wikipedia.org/wiki/Accuracy_and_precision

without limits. How can this be done? One simple way is to put an upper limit to the number of Monte Carlo runs. If the estimate of the input parameter μ_X is based on a sample of $n = 16$ data points, perhaps we should not do more than $N_{\text{run}} = 16$ Monte Carlo runs. While this sounds fair, a complication is that we need more guidance on the case of more complicated functions than just $Y = X$, for instance $Y = X_1 X_2^2 + \frac{\ln X_1}{4+\sin X_2}$. If X_1 has been sampled with $n_{X_1} = 16$ and X_2 with $n_{X_2} = 9$, what should we take for the number of Monte Carlo runs, N_{run}? Perhaps the weakest link defines our maximum quality, so our Monte Carlo run could do with just 9 runs in this case. The result is a very imprecise estimate of μ_Y, but visibly imprecise. The solution of taking a small number of Monte Carlo runs by the way also solves the problem of overly significant results (Heijungs et al. 2016).

Another remedy is of course to determine the parameters of the input distributions with more precision, so using a larger sample size n_{X_1}, n_{X_2}, etc. In practice, this is, however, not easy. Many of the millions of data in the LCA model come from general purpose generic databases, and recollecting these data from multiple sites and at multiple days would be a horrendous task.

A final point is the case of probability distributions with parameters that have not been estimated from data, but for which a procedural estimation has been used. An important example is the earlier-mentioned pedigree approach, where data quality indicators, for instance for representativeness and age, define default standard deviations. The popular ecoinvent database is a major example here (Frischknecht et al. 2004; Weidema et al. 2013), but the approach is also becoming popular in other areas (Laner et al. 2016). For such data, it is often unclear what the sample size of the data is, so it is not possible to estimate the precision of the mean in terms of a standard error. But, it will be clear that the parameters of the input distribution are not at all accurate, so a propagation into almost infinitely precise Monte Carlo output results is as misleading as the parameter-based procedure on which our main argument was based. An ultimate consequence is that such pedigree-based probability distributions are incompatible with large-scale Monte Carlo simulations. This is an important take-home message of our analysis, because the pedigree approach has grown into a major paradigm for estimating standard deviations of LCA data, and Monte Carlo has become the default procedure for propagating uncertainties in LCA. The incompatibility of the two has, as far we know, not been recognized before, and our analysis does not suggest any way out. This suggests a major area of research in dealing with uncertainty in LCA.

References

BIPM (Bureau International des Poids et Mesures) (2008) Evaluation of measurement data – Supplement 1 to the "Guide to the expression of uncertainty in measurement" – Propagation of distributions using a Monte Carlo method. (https://www.bipm.org/utils/common/documents/jcgm/JCGM_101_2008_E.pdf)

Burmaster DE, Anderson PD (1994) Principles of good practice for the use of Monte Carlo techniques in human health and ecological risk assessments. Risk Anal 14:477–481

Casella R, Berger RL (2002) Statistical inference. Second edition, Duxbury

Frischknecht R, Jungbluth N, Althaus H-J, Doka G, Heck T, Hellweg S, Hischier R, Nemecek T, Rebitzer G, Spielmann M (2004) Overview and methodology. Ecoinvent report no. 1. Swiss Centre for Life Cycle Inventories

Gharamani S (2005) Fundamentals of probability with stochastic processes. Third edition, Pearson

Helton JC, Johnson JD, Sallaberry CJ, Storlie CB (2006) Survey of sampling-based methods for uncertainty and sensitivity analysis. Rel Eng Sys Saf 91:1175–1209

Heijungs R, Suh S (2002) The computational structure of life cycle assessment. Kluwer Academic Publishers

Heijungs R, de Koning A, Wegener Sleeswijk A (2015) Sustainability analysis and systems of linear equations in the era of data abundance. J Env Acc Man 3:109–122

Heijungs R, Henriksson PJG, Guinée JB (2016) Measures of difference and significance in the era of computer simulations, meta-analysis, and big data. Entropy 18:361

Heijungs R, Guinée JB, Mendoza Beltrán A, Henriksson PJG, Groen E (2019) Everything is relative and nothing is certain. Toward a theory and practice of comparative probabilistic LCA. Int J Life Cycle Assess 24:1573–1579

Hongxiang C, Wei C (2013) Uncertainty analysis by Monte Carlo simulation in a life cycle assessment of water-saving project in green buildings. Inf Technol J 12:2593–2598

Huijbregts MAJ (1998) Application of uncertainty and variability in LCA. Part I: a general framework for the analysis of uncertainty and variability in life cycle assessment. Int J Life Cycle Assess 3: 273–280

ISO (2006) ISO 14044. Environmental Management – Life Cycle Assessment – Requirements and Guidelines. International Organization for Standardization

JRC-IES (2010) ILCD Handbook. International Reference Life Cycle Data System. General Guide for Life Cycle Assessment. Joint Research Centre

Laner D, Feketitsch J, Rechberger H, Fellner J (2016) A novel approach to characterize data uncertainty in material flow analysis and its application to plastics flows in Austria. J Ind Ecol 20:1050–1063

Lloyd SM, Ries R (2007) Characterizing, propagating, and analyzing uncertainty in life-cycle assessment. J Ind Ecol 11:161–181

McCleese DL, LaPuma PT (2002) Using Monte Carlo simulation in life cycle assessment for electric and internal combustion vehicles. Int J Life Cycle Assess 7:230–236

Mendoza Beltrán MA, Prado V, Font Vivanco D, Henriksson PJG, Guinée JB, Heijungs R (2018) Quantified uncertainties in comparative life cycle assessment: what can be concluded? Environ Sci Technol 52:2152–2161

Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Ass 44:335–341

Muller S, Lesage P, Ciroth A, Mutel C, Weidema BP, Samson R (2016) The application of the pedigree approach to the distributions foreseen in ecoinvent v3. Int J Life Cycle Assess 21:1327–1337

Rice JA (2007) Mathematical statistics and data analysis. Third edition, Thomson

Ross S (2010) A first course in probability. Eighth edition, Pearson

Shonkwiler RW, Mendivil F (2009) Explorations in Monte Carlo methods. Springer

Weidema BP, Bauer C, Hischier R, Mutel C, Nemecek T, Reinhard J, Vadenbo CO, Wernet G (2013) Overview and methodology. Data quality guideline for the ecoinvent database version 3. Ecoinvent Report 1 (v3). The ecoinvent Centre

Xin L (2006) Uncertainty and sensitivity analysis of a simplified ORWARE model for Jakarta. Stockholm (https://www.diva-portal.org/smash/get/diva2:411539/FULLTEXT01.pdf)