Gusmão *et al. J Egypt Math Soc* (2021) 29:21
https://doi.org/10.1186/s42787-021-00130-x

Journal of the Egyptian Mathematical Society

ORIGINAL RESEARCH

Open Access

Analyzing and solving the identifiability problem in the exponentiated generalized Weibull distribution

Felipe R. S. de Gusmão[1], Frank Gomes-Silva[1*], Cícero C. R. de Brito[2], Fábio V. J. Silveira[1], Jader S. Jale[1], Sílvio F. A. Xavier-Júnior[3] and Pedro R. D. Marinho[4]

*Correspondence: franksinatrags@gmail.com
[1] DEINFO, Universidade Federal Rural de Pernambuco, Recife, Brazil
Full list of author information is available at the end of the article

Abstract
The well-known Weibull distribution can be used to model the decreasing and unimodal failure rate quite standard in reliability and biological studies. It is also commonly adopted as baseline to generate new distributions from generalized classes. In this paper, we investigate the identifiability of the exponentiated generalized class of distributions and in particular the exponentiated generalized Weibull distribution. We also develop conditions under which the model becomes identifiable. To further illustrate the identifiability issue, we consider a simulation study, and an application is presented to illustrate the potentialities of the model with the new parameterization.

Keywords: Exponentiated, Generalized, Weibull, Identifiability, Simulation, Application

Mathematics Subject Classification: 62F10, 97K50, 97K60

Introduction

Lately, many authors have proposed new classes of distributions, which are modifications of the cumulative distribution functions (cdf) that provide hazard rate functions (hrf) taking various shapes. We can cite the exponentiated Weibull ($\mathcal{EW}$)[1, 21, 22], which has an upside-down bathtub (unimodal) hrf form [2]. Carrasco et al. [3] showed a four-parameter distribution denoted generalized modified Weibull distribution whose hrf exhibits non-monotonic shapes such as a bathtub and upside-down bathtub; Gusmão et al. [4] introduced and studied the tri-parametric inverse Weibull generalized distribution that possesses failure rate with unimodal, increasing and decreasing form.

Several families proposed in the literature comprise a source of probability distributions for modeling lifetime data, since, in general, the resulting distribution and the baseline have the same support. Cordeiro et al. [5] proposed a new family, the exponentiated generalized ($\mathcal{EG}$) class of distributions, to generalize other distributions. Considering that a random variable T has distribution G, they suggest applying the new class of distributions to generalize any distribution G by

$$F_G(t;a,b) = \left\{1 - [1 - G(t)]^a\right\}^b, \tag{1}$$

where $a > 0$ and $b > 0$ are two additional shape parameters. The authors point out that the new class of distributions is simpler and more tractable than the generalized beta family [6]. The quantile function (qf) of the new class has closed form. It entails that simulations regarding (1) are easier to perform.

The following well-known baseline distributions have been discussed in recent works for the exponentiated generalized class [5] (this list is not exhaustive): Birnbaum–Saunders distribution [7], generalized gamma distribution [8], Gumbel distribution [9], Dagum distribution [10], Weibull distribution [11], extended exponential distribution [12], arcsine distribution [13], standardized half-logistic distribution [14], extended Pareto distribution [15] standardized Gumbel distribution [16] and extended Gompertz [17].

It is well-known that the addition of parameters to distribution classes can lead to identifiability problems and consequently bring complications to the estimation of parameters in the proposed model. According to [18], a parameter $\boldsymbol{\theta}$ for a family of distributions $\left\{f(x,\boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\right\}$ is identifiable if different values of $\boldsymbol{\theta}$ correspond to different probability density functions (pdf) or probability mass functions. That is, if $\boldsymbol{\theta} \neq \boldsymbol{\theta'}$, then $f(x,\boldsymbol{\theta}) \neq f\left(x,\boldsymbol{\theta'}\right)$.

Jones et al. [19] define identifiability as follows: Consider a stack of probabilities $p_1, ..., p_n$, $n \in \mathbb{N}$, within a single vector $\boldsymbol{\psi}$ with dimensions $q \times 1$ and the parametric model with a vector $\boldsymbol{\gamma}$ with dimensions $r \times 1$. The presented model, implicitly specifies, a function F that determines how $\boldsymbol{\psi}$ is calculated from $\boldsymbol{\gamma}$,

$$\boldsymbol{\psi} = F(\boldsymbol{\gamma}).$$

Hence, the model will be identifiable if F is an invertible function; it follows that there is a one-to-one correspondence between $\boldsymbol{\gamma}$ and $\boldsymbol{\psi}$. If $\boldsymbol{\gamma}_1 \neq \boldsymbol{\gamma}_2$ and $F\left(\boldsymbol{\gamma}_1\right) = F\left(\boldsymbol{\gamma}_2\right)$, the model will have identifiability problems. Nevertheless, Jones et al. [19] state that the model will be locally identifiable in a particular $\boldsymbol{\gamma}$ if F is an invertible function in the vicinity of $\boldsymbol{\gamma}$.

In a review paper on statistical identifiability, Paulino and Pereira [20] studied issues like parallelism between parametric identifiability and sample sufficiency. They also discussed how identifiability, measures of sample information and inferential estimation concepts are related. Additionally, classic and Bayesian methods were considered as strategies for making inferences on models with parametric identification problems.

Based on the aforementioned ideas and considering the relation between the parameters of the exponentiated generalized class of distributions and the baseline function, we used the Weibull distribution as a candidate for G. Using Eq. (1) and performing some mathematical manipulations, we obtain a parameterization for the exponentiated generalized Weibull ($\mathcal{EGW}$) distribution that was introduced by [11]. It was also studied by [1, 21, 22]. This paper aims to study the similarities that evince the problem of identifiability of the $\mathcal{EGW}$ distribution.

Methods

The $\mathcal{EGW}$ distribution and a study on identifiability

The Weibull distribution has received considerable attention in the statistical literature. Many authors have studied the shapes of the density and failure rate functions for the basic model of the Weibull distribution. Let T be a random variable with Weibull distribution, then its cdf can be written as:

$$G(t) = 1 - \exp\left[-(\alpha t)^{\beta}\right], \quad t > 0, \tag{2}$$

where $\alpha > 0, \beta > 0$.

Replacing $G(t)$ in Eq. (1) by (2), we have

$$F_{\mathcal{EGW}}(t; a, b, \alpha, \beta) = \left\{1 - \exp\left[-a(\alpha t)^{\beta}\right]\right\}^{b} \tag{3}$$

where $F_{\mathcal{EGW}}(\cdot)$ is the $\mathcal{EGW}$ cdf. The pdf is given by

$$f_{\mathcal{EGW}}(t; a, b, \alpha, \beta) = a\, b\, \beta\, \alpha^{\beta}\, t^{\beta-1} \exp\left[-a(\alpha t)^{\beta}\right]\left\{1 - \exp\left[-a(\alpha t)^{\beta}\right]\right\}^{b-1}, \tag{4}$$

where $\boldsymbol{\theta} = (a, b, \alpha, \beta)$ is the vector of parameters of $F_{\mathcal{EGW}}(t; a, b, \alpha, \beta)$.

Consider that $\boldsymbol{\Theta}_{\mathcal{EGW}}$ is the parametric space of the $\mathcal{EGW}$ distribution, Γ is a specific set of indices and $\boldsymbol{\theta}_i = (a_i, b_i, \alpha_i, \beta_i) \in \boldsymbol{\Theta}_{\mathcal{EGW}}$ where $a_i, b_i, \alpha_i, \beta_i > 0$ for all $i \in \Gamma$. Let $F_{\boldsymbol{\Theta}_{\mathcal{EGW}}} = \{F_{\mathcal{EGW}}(t; \boldsymbol{\theta}_i) : \boldsymbol{\theta}_i \in \boldsymbol{\Theta}_{\mathcal{EGW}}, \forall i \in \Gamma\}$ be a family of cdfs of the $\mathcal{EGW}$ distribution. Given that $i \neq j$ for all $i, j \in \Gamma$, if $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j \Rightarrow F_{\mathcal{EGW}}(t; \boldsymbol{\theta}_i) = F_{\mathcal{EGW}}\left(t; \boldsymbol{\theta}_j\right)$, we say that $\boldsymbol{\Theta}_{\mathcal{EGW}}$ is not identifiable.

Let $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ be such that $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j$ with $a_i \neq a_j$, $b_i = b_j = b$, $\alpha_i \neq \alpha_j$ and $\beta_i = \beta_j = \beta$. Then, by hypothesis, we have that

$$\alpha_i \neq \alpha_j \Rightarrow (\alpha_i t)^{\beta} \neq \left(\alpha_j t\right)^{\beta}.$$

Take $a_i = \frac{a_j \alpha_j^{\beta}}{\alpha_i^{\beta}}$, then

$$\exists \quad a_i \neq a_j : a_i(\alpha_i t)^{\beta} = a_j\left(\alpha_j t\right)^{\beta} \Rightarrow F_{\mathcal{EGW}}(t; \boldsymbol{\theta}_i) = F_{\mathcal{EGW}}\left(t; \boldsymbol{\theta}_j\right).$$

Therefore, the $\boldsymbol{\Theta}_{\mathcal{EGW}}$ is not identifiable.

The $\mathcal{EW}$ distribution and a study on identifiability

The reparameterization performed on the parameters $\alpha a^{\frac{1}{\beta}}$ solves the problem of identifiability, see the work of [23], where a is the parameter recently introduced. Without this reparameterization various values of a and α satisfy the relation $c = a\alpha^{\beta}$ for fixed value of c. With the cited relation it is possible to rewrite Eq. (3), obtaining the $\mathcal{EW}$ cdf:

$$F_{\mathcal{EW}}(t; b, c, \beta) = \left\{1 - \exp\left[-(ct)^{\beta}\right]\right\}^{b}, \tag{5}$$

wherein $b > 0$ is the shape parameter, and $c > 0$ is the scale parameter. Hence, the $\mathcal{EW}$ distribution has three parameters, and its pdf is given by

$$f_{\mathcal{EW}}(t; b, c, \beta) = \beta\, b\, c^{\beta}\, t^{(\beta-1)} \exp\left[-(ct)^{\beta}\right]\left\{1 - \exp\left[-(ct)^{\beta}\right]\right\}^{b-1}. \tag{6}$$

Consider that $\boldsymbol{\Theta}_{\mathcal{EW}}$ is the parametric space of the $\mathcal{EW}$ distribution, Γ is a specific set of indices and $\boldsymbol{\theta}_i = (b_i, c_i, \beta_i) \in \boldsymbol{\Theta}_{\mathcal{EW}}$ where $b_i, c_i, \beta_i > 0$ for all $i \in \Gamma$. Let $F_{\boldsymbol{\Theta}_{\mathcal{EW}}} = \{F_{\mathcal{EW}}(t; \boldsymbol{\theta}_i) : \boldsymbol{\theta}_i \in \boldsymbol{\Theta}_{\mathcal{EW}}, \forall i \in \Gamma\}$ be a family of cdfs of the $\mathcal{EW}$ distribution.

Given that $i \neq j$ for all $i, j \in \Gamma$, if $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j \Rightarrow F_{\mathcal{EW}}(t; \boldsymbol{\theta}_i) = F_{\mathcal{EW}}\left(t; \boldsymbol{\theta}_j\right)$, we say that $\boldsymbol{\Theta}_{\mathcal{EW}}$ is not identifiable.

The vector $\boldsymbol{\theta}_i$ differs from $\boldsymbol{\theta}_j$ in seven ways. Next, consider *Case 1*. Let $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ such that $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j$ with $b_i \neq b_j$, $c_i = c_j = c$ and $\beta_i = \beta_j = \beta$. Then, from this hypothesis, we have the following chain of implications:

$$\begin{aligned} b_i \neq b_j &\Rightarrow \left\{1 - \exp\left[-(ct)^{\beta}\right]\right\}^{b_i} \neq \left\{1 - \exp\left[-(ct)^{\beta}\right]\right\}^{b_j} \\ &\Rightarrow F_{\mathcal{EW}}(t; \boldsymbol{\theta}_i) \neq F_{\mathcal{EW}}\left(t; \boldsymbol{\theta}_j\right). \end{aligned}$$

Table 1 summarizes the proof of identifiability for each of the other cases from the hypothesis, and also displays its appropriate implications.

Therefore, the $\boldsymbol{\Theta}_{\mathcal{EW}}$ is identifiable.

Note that $F_{\mathcal{EGW}}$ and $F_{\mathcal{EW}}$ are equal functions, as long as they have the same domain and image set. However, $F_{\mathcal{EW}}$ as an identifiable cdf has reliable estimation which is quite different from $F_{\mathcal{EGW}}$. Let $F_{\mathcal{EGW}}(t; \boldsymbol{\theta})$ for all $t > 0$ and $\boldsymbol{\theta} = (a, b, \alpha, \beta)$. Hence,

$$F_{\mathcal{EGW}}(t; \boldsymbol{\theta}) = \left\{1 - \exp\left[-a(\alpha t)^{\beta}\right]\right\}^{b} = \left\{1 - \exp\left[-a\alpha^{\beta} t^{\beta}\right]\right\}^{b}.$$

Let $c^{\beta} = a\alpha^{\beta}$ where $c > 0$, hence we have that

$$F_{\mathcal{EGW}}(t; \boldsymbol{\theta}) = \left\{1 - \exp\left[-c^{\beta} t^{\beta}\right]\right\}^{b} = \left\{1 - \exp\left[-(ct)^{\beta}\right]\right\}^{b} = F_{\mathcal{EW}}\left(t; \boldsymbol{\theta}'\right)$$

where $\boldsymbol{\theta}' = (b, c, \beta)$.

Therefore, $F_{\mathcal{EGW}}(t; \boldsymbol{\theta}) = F_{\mathcal{EW}}\left(t; \boldsymbol{\theta}'\right)$ for all $t > 0$.

Results and discussion

Monte Carlo simulations based on $\mathcal{EGW}$ and $\mathcal{EW}$ models

Computational experiments play an important role in probability and statistics since they can verify the validity of a hypothesis, examine the performance of something new or demonstrate a known truth. In this section, we present the estimates of the parameters under the maximum likelihood method for the $\mathcal{EGW}$ and $\mathcal{EW}$ models. They were obtained via BFGS, SANN, and Nelder–Mead, implemented in R `OPTIM` function [24]. For this, we implemented two other functions to automate the simulations: *fitDist* and *getSimulation*. The pseudo-codes of those algorithms as well as

Table 1 Proof that $\boldsymbol{\Theta}_{\mathcal{EW}}$ is identifiable

Cases	*Hypothesis*: $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j$	*Implication for the thesis*
1	$b_i \neq b_j, c_i = c_j$ and $\beta_i = \beta_j$	$F_{EW}(t; \boldsymbol{\theta}_i) \neq F_{EW}\left(t; \boldsymbol{\theta}_j\right)$
2	$b_i = b_j, c_i = c_j = c$ and $\beta_i \neq \beta_j$	$\beta_i \neq \beta_j \Rightarrow (ct)^{\beta_i} \neq (ct)^{\beta_j}$
3	$b_i = b_j, c_i \neq c_j$ and $\beta_i = \beta_j = \beta$	$c_i \neq c_j \Rightarrow (c_i t)^{\beta} \neq \left(c_j t\right)^{\beta}$
4	$b_i = b_j, c_i \neq c_j$ and $\beta_i \neq \beta_j$	$c_i \neq c_j \Rightarrow (c_i t)^{\beta_i} \neq \left(c_j t\right)^{\beta_j}$
5	$b_i \neq b_j, c_i \neq c_j$ and $\beta_i = \beta_j = \beta$	$c_i \neq c_j \Rightarrow (c_i t)^{\beta} \neq \left(c_j t\right)^{\beta}$
6	$b_i \neq b_j, c_i = c_j = c$ and $\beta_i \neq \beta_j$	$c_i = c_j \Rightarrow (ct)^{\beta_i} \neq (ct)^{\beta_j}$
7	$b_i \neq b_j, c_i \neq c_j$ and $\beta_i \neq \beta_j$	$c_i \neq c_j \Rightarrow (c_i t)^{\beta_i} \neq \left(c_j t\right)^{\beta_j}$

these functions can be seen in "Appendix." Nowadays, with the available computational resources, such as parallel processing of many cores and multiple processes, it is possible speed-up the results of the computational simulations. Therefore, we run the simulations on parallel processes to explore the high-performance computing and runtime optimization. Thus, the results of the simulations as well as their execution times were gathered from a notebook Intel® Core™ i5-7200U, CPU 2.50 GHz, 2712 Mhz, 2 cores, 4 logical processors, RAM 8.00 GB, Microsoft® Windows 10 Home Single Language, X64 system, R© version 3.6.1, and RStudio© version 1.2.5001.

Simulation for the $\mathcal{EGW}$ distribution

Samples of size $50,\ 100,\ 500\ \text{ and }\ 1000$ were obtained using the $\mathcal{EGW}$ qf given by

$$Q_{\mathcal{EGW}}(q) = \left\{ \log \left[1 - q^{\frac{1}{b}} \right]^{-\left(\frac{1}{a\alpha^{\beta}}\right)} \right\}^{\frac{1}{\beta}}, \tag{7}$$

where q takes random values from a $U(0,1)$, adopting $a = 2$, $b = 3$, $\alpha = 4$ and $\beta = 5$. The estimates were acquired by the maximum likelihood method via BFGS, SANN, and Nelder–Mead.

Figures 1 and 2 display the histogram from simulated data of the $\mathcal{EGW}$ distribution with density for the $\mathcal{EGW}$ distribution and the empirical distribution for data set size of $50,\ 100,\ 500\ \text{ and }\ 1000$. The histogram was obtained using the qf of the $\mathcal{EGW}$ distribution, and the algorithms BFGS, SANN, and Nelder–Mead obtained estimates via MLE.

Next, we present the results of the parameter estimation using the $\mathcal{EGW}$ distribution. The BFGS method for estimating parameter a proved to be inefficient, even with the increase in the number of simulated data. For parameter b, the estimates showed reasonable results for 500 and 1000 simulated data. However, the method was not

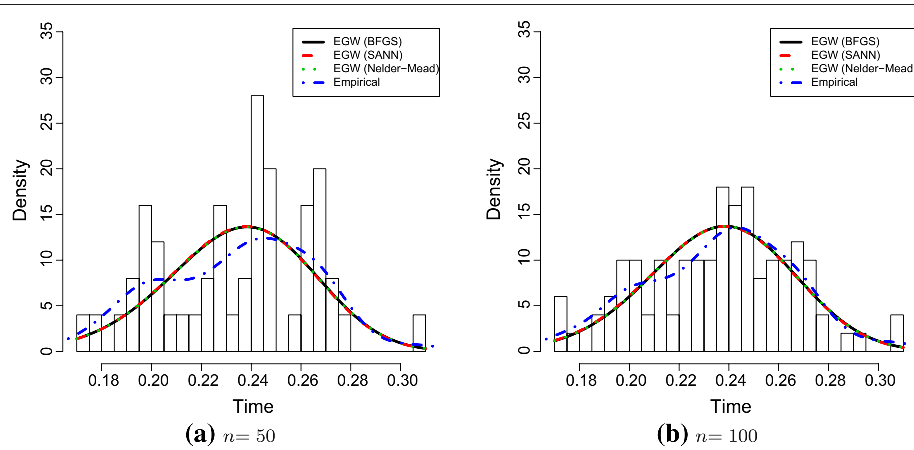


Fig. 1 Estimated densities for $\mathcal{EGW}$ distribution and the distribution of the empirical values for the sets of simulated data of sizes 50 and 100

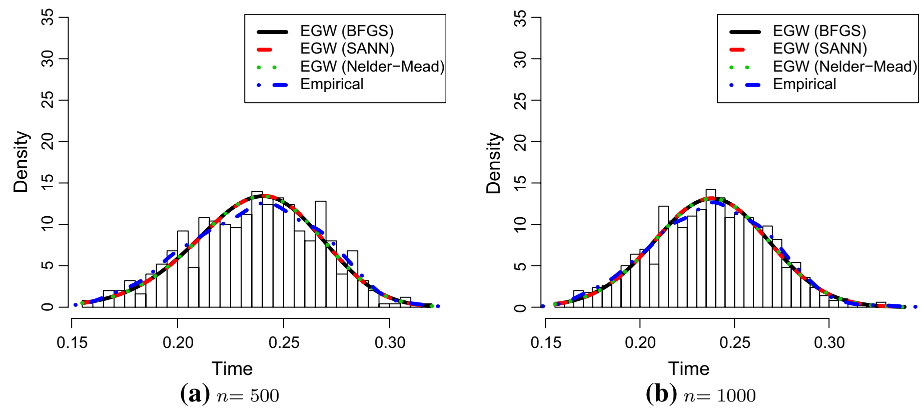


Fig. 2 Estimated densities for $\mathcal{EGW}$ distribution and the distribution of the empirical values for the sets of simulated data of sizes 500 and 1000

satisfactory regarding the α parameter. Finally, a reasonable result was obtained for the β parameter only for 1000 simulated data.

Regarding the SANN method, the estimation was inefficient for the parameters a and α. The estimates for parameter b were reasonable only from 500 simulated data. For the β parameter, there was a reasonable estimate only when 1000 simulated data was reached.

The Nelder–Mead method did not give satisfactory results for the estimation of parameters a and α. However, it presented a reasonable estimate for parameter b from 500 simulated data, as well as for the β parameter, but only for 1000 simulated data.

In the simulations concerning the estimation of the parameters of the $\mathcal{EGW}$ distribution, we obtained 81.25% (39/48) of inefficient estimates, 18.75% (9/48) of reasonable estimates and none satisfactory.

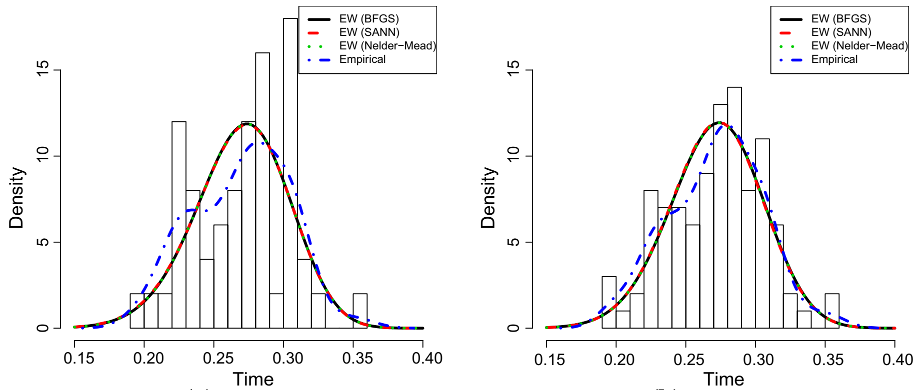
The graphs of all methods showed equivalent adjustments; more details are available in "Appendix." See Table 2 including the standard error (SE) and the mean squared error (MSE) and Figs. 1, 2.

Simulation for $\mathcal{EW}$ distribution

Although it is a well-known model and numerous other models generalize it, to our knowledge, simulation studies have not been carried out with the $\mathcal{EW}$ distribution. Samples of size 50, 100, 500, and 1000 were obtained using the qf of the $\mathcal{EW}$ distribution. The results of the simulations are presented in Table 3. The $\mathcal{EW}$ qf is given by

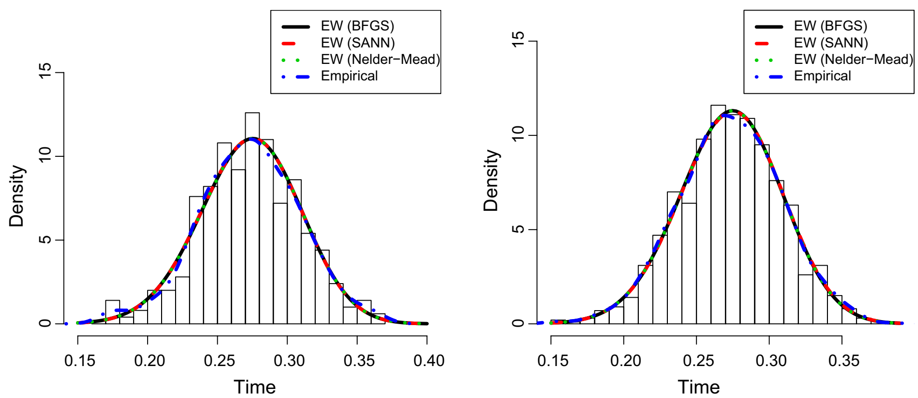
$$Q_{EW}(q) = \left\{ \log \left[1 - q^{\frac{1}{b}} \right]^{-\left(\frac{1}{c^{\beta}}\right)} \right\}^{\frac{1}{\beta}}, \tag{8}$$

where q takes random values from a $U(0,1)$ adopting $b = 3$, $c = 4$, and $\beta = 5$. We obtain points of the $\mathcal{EW}$ distribution given by (8).



(a) $n = 50$ **(b)** $n = 100$

Fig. 3 Estimated densities for $\mathcal{EW}$ distribution and the distribution of the empirical values for the sets of simulated data of sizes 50 and 100



(a) $n = 500$ **(b)** $n = 1000$

Fig. 4 Estimated densities for $\mathcal{EW}$ distribution and the distribution of the empirical values for the sets of simulated data of sizes 500 and 1000

Figures 3 and 4 present the histogram from simulated data of the $\mathcal{EW}$ distribution density and the empirical distribution for data size of 50, 100, 500, and 1000 using the $\mathcal{EW}$ qf, and BFGS, SANN, and Nelder–Mead performed the estimates via MLE.

The estimation of the parameters of the $\mathcal{EW}$ distribution presented the following results.

For the BFGS method, with only 1000 simulated data, there was a reasonable result in estimating parameter b. Regarding parameter c, with 500 simulated data, we observed a reasonable estimate. However, for 1000 observations, the BFGS method had a satisfactory result. Regarding the β parameter, the estimates were reasonable only from 500 simulated data.

With respect to the SANN method, the estimates for parameter b were reasonable only for 1000 simulated data. For parameter c, there was a reasonable estimate for 500 simulated data. However, for 1000 simulated data, the estimation was satisfactory. For 500 simulated data onwards, the β parameter estimates were reasonable.

Finally, for the Nelder–Mead method, the estimation of parameter b was reasonable only for 1000 simulated data. The estimates for parameter c were reasonable and satisfactory, for 500 and 1000 simulated data, respectively. From 500 simulated data, the estimates for the β parameter were reasonable.

For the simulations generated for the $\mathcal{EW}$ distribution, we obtained 58.33% (21/36) inefficient estimates, 33.34% (12/36) reasonable and 8.33% (3/36) satisfactory.

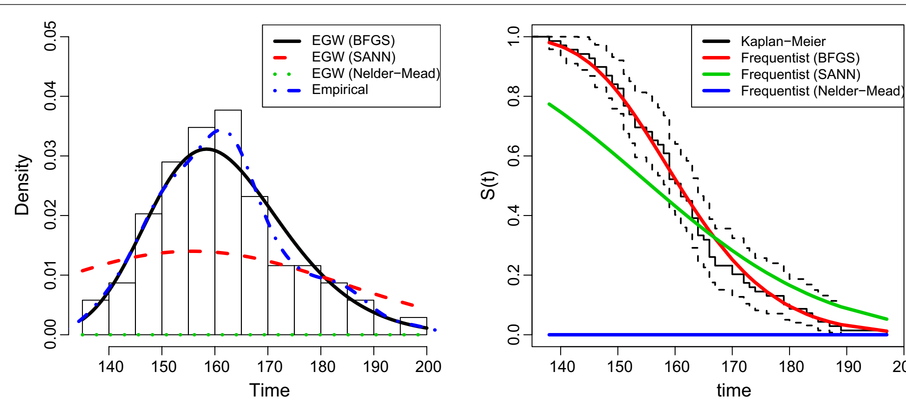
Thus, we can observe that the identifiability (reparameterization) of the $\mathcal{EW}$ distribution provided better results in the simulations, as it decreased the amount of inefficient estimates (81.25% → 58.33%) and increased the amount of reasonable estimates (18.75% → 33.34%) and satisfactory (0% → 8.33%).

The ratio between the execution times (in seconds) of the simulations of the $\mathcal{EGW}$ and $\mathcal{EW}$ distributions were as follows: 61,052/31845 (1.92), 164,702/55,106 (2.99), 397,079/231,317 (1.72), and 590,006/390,454 (1.51). These results show that the $\mathcal{EW}$ distribution requires a much shorter execution time. Thus, the identifiability of the $\mathcal{EW}$ distribution has the additional advantage of optimizing the time for running computer simulations.

Application with the $\mathcal{EGW}$ distribution and the $\mathcal{EW}$ distribution

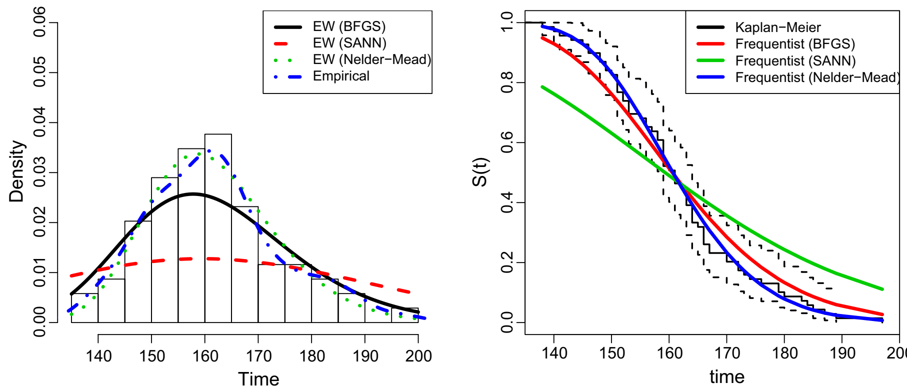
In this section, we analyze a real data set of Nelore cattle [25] using the $\mathcal{EGW}$ distribution and the $\mathcal{EW}$ distribution. The algorithms of BFGS, SANN, and Nelder–Mead performed the maximum likelihood estimates. The commercial production of beef in Brazil, which mostly originates from the Nelore breed, searches to optimize the process to obtain a time for the calves to reach the specific weight from their birth to weaning. We observed the data with 69 Nelore bulls, the time (in days) until the animals achieved the weight of 160kg relative to the period from birth to weaning.

Figure 5 exhibits the results obtained for $\mathcal{EGW}$ such as the plot 5 and the parameters estimation table (Table 4 in "Appendix"). One can note that the BFGS method performed a better fit concerning the empirical function and to the histogram than the other methods proposed in this article.



(a) Estimated density **(b)** Estimated survival function

Fig. 5 **a** Estimated density for $\mathcal{EGW}$ distribution and the empirical distribution for the set of Nelore data. **b** Estimated survival function for $\mathcal{EGW}$ distribution and the Kaplan–Meier distribution for the set of Nelore data with a confidence interval of 0.95



(a) Estimated density **(b)** Estimated survival function

Fig. 6 **a** Estimated density for $\mathcal{EW}$ distribution and the empirical distribution for the set of Nelore data. **b** Estimated survive function for $\mathcal{EW}$ distribution and the Kaplan–Meier distribution for the set of Nelore data with a confidence interval of 0.95

Analyzing the plots in Fig. 6 and the results Tables (see the Table 5 in "Appendix"), it is observed that the Nelder–Mead method adjusted the $\mathcal{EW}$ distribution, concerning the histogram and the empirical function, better than the other methods. Notwithstanding, the estimation of the parameters by the Nelder–Mead method did not produce results for the SE of parameters b and c. Hence, as the estimation of the parameters by the BFGS method was the second-best fit, and the results were also produced for the SE for the parameters b, c and β one can consider that the BFGS method performed the most suitable adjustment for the data via $\mathcal{EW}$ distribution.

Table 4 (in "Appendix") shows that the Nelder–Mead method was able to perform the estimation of the parameters of the $\mathcal{EGW}$ distribution, but there was failure to report the SE, since the produced Hessian returned NaN (abbreviation for Not a Number) for the first row and the first column, whose information refers to the parameter a.

This suggests that the solution found by the Nelder–Mead method is not reliable, in this case, and consequently, that the model adjusted by the estimates of the parameters found is not suitable for these data. This fact may be related to the lack of identifiability of the $\mathcal{EGW}$ distribution.

Conclusions

In this study, we presented a technique to reduce the parameters of the exponentiated generalized Weibull distribution ($\mathcal{EGW}$). Additionally, we identified that the exponentiated Weibull distribution ($\mathcal{EW}$) displayed more parsimony and identifiability in the parameters than the $\mathcal{EGW}$. The performances of the two distributions were analyzed using simulated and a real dataset; the $\mathcal{EW}$ performed slightly better with simulated data and lightly worse with real data.

Appendix

See Tables 2, 3, 4 and 5.

Table 2 MLE estimates for the parameters of $\mathcal{EGW}$ distribution with simulated data from $\mathcal{EGW}$ distribution via `BFGS`, `SANN`, and `Nelder−Mead` algorithms

n	Method	Inference results	a	b	α	β
50	BFGS	Estimates	6.559792	4.508665	3.222132	5.679618
		SE	21.726748	5.959038	142.025577	2.451735
		MSE	21.252218	30.410082	0.624046	5.544654
	SANN	Estimates	5.758308	4.205869	3.638858	5.638019
		SE	5.773811	5.391740	1.419861	2.597639
		MSE	36.580530	17.459904	0.942536	4.807788
	Nelder–Mead	Estimates	2.445578	4.636861	4.541140	5.677241
		SE	34.326846	6.496251	71.788950	2.461764
		MSE	10.188337	39.203020	2.132110	5.576713
Time	0d:16h:57m:32s (61052 s)					
100	BFGS	Estimates	10.076350	3.687458	2.938870	5.431711
		SE	3.324494	2.853083	7.095580	1.568548
		MSE	66.070093	9.705820	1.132114	2.587435
	SANN	Estimates	5.860451	3.682335	3.611025	5.428296
		SE	5.255925	2.956691	1.248130	1.629309
		MSE	38.041064	9.051131	0.890948	2.579751
	Nelder–Mead	Estimates	1.833553	3.704633	4.466173	5.430765
		SE	29.335765	2.898004	27.704366	1.570119
		MSE	3.074604	10.360465	1.047837	2.595298
Time	1d:21h:45m:2s (164702 s)					
500	BFGS	Estimates	10.228551	3.052343	2.898200	

BFGS algorithm

Henceforward, the following notation is used: p is the number of parameters to be estimated, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^\top \in \Theta$ is the vector of unknown parameters, $\boldsymbol{\theta_0} = (\theta_{01}, \ldots, \theta_{0p})^\top \in \Theta$ the initial guess solution, f the objective function (minimization by default) representing the log-likelihood function $\ell(\boldsymbol{\theta}|x)$, and x the dataset. The BFGS is a Quasi-Newton second-derivative line search method used to solve unconstrained optimization problems. Algorithm 1 shows the pseudo-code of the BFGS algorithm [26–29].

Table 3 MLE estimates for the parameters of $\mathcal{EW}$ distribution with simulated data from $\mathcal{EW}$ distribution via `BFGS`, `SANN`, and `Nelder–Mead` algorithms

n	Method	Inference results	b	c	β
50	BFGS	Estimates	4.402925	4.102611	5.949618
		SE	5.697425	0.751590	2.601159
		MSE	22.795855	0.402740	9.699106
	SANN	Estimates	4.449711	4.106828	5.920183
		SE	6.085270	0.789521	2.686872
		MSE	24.098193	0.413645	8.739624
	Nelder–Mead	Estimates	4.472318	4.107355	5.951902
		SE	5.962889	0.764355	2.603384
		MSE	25.399142	0.420800	10.046063
Time	0d:8h:50m:45s (31845 s)				
100	BFGS	Estimates	4.213534	4.113357	5.299846
		SE	3.566710	0.509471	1.548855
		MSE	18.221747	0.290249	2.756462
	SANN	Estimates	4.207505	4.114013	5.297310
		SE	3.650426	0.521174	1.571037
		MSE	17.379728	0.288609	2.753479
	Nelder–Mead	Estimates	4.249701	4.115646	5.298766
		SE	3.666593	0.513693	1.549940
		MSE	19.982132	0.299304	2.763964
Time	0d:15h:18m:26s (55106 s)				
500	BFGS	Estimates	3.155945	4.017774	5.055031
		SE	0.906044	0.185317	0.627086
		MSE	0.906766	0.035019	0.396619
	SANN	Estimates	3.157440	4.017996	5.054715
		SE	0.911430	0.186253	0.629546
		MSE	0.911503	0.035196	0.398176
	Nelder–Mead	Estimates	3.156035	4.017759	5.055316
		SE	0.906550	0.185396	0.627336
		MSE	0.908748	0.035088	0.397327
Time	2d:16h:15m:17s (231317 s)				
1000	BFGS	Estimates	3.078456	4.009790	5.023386
		SE	0.609515	0.128449	0.437921
		MSE	0.390373	0.016572	0.189574
	SANN	Estimates	3.078632	4.009775	5.023852
		SE	0.611345	0.128798	0.439047
		MSE	0.393664	0.016703	0.190998
	Nelder–Mead	Estimates	3.077648	4.009598	5.024238
		SE	0.609420	0.128441	0.438062
		MSE	0.391118	0.016610	0.190232
Time	4d:12h:27m:34s (390454 s)				

Table 4 MLE estimates for the parameters of $\mathcal{EGW}$ distribution with Nelore data via `BFGS`, `SANN`, and `Nelder-Mead` algorithms

Methods	Parameters	Estimates	SE	Confidence Interval (0.95)
BFGS	a	0.01631688	0.004142414	[0.01625805; 0.01637572]
	b	58.74132600	27.050302156	[58.35713331; 59.12551870]
	α	0.03633065	0.004862259	[0.03626159; 0.03639970]
	β	3.17753052	0.241423958	[3.17410160; 3.18095944]
SANN	a	0.801092593	0.495883447	[0.794049611; 0.808135576]
	b	4.477365906	0.822148967	[4.465689008; 4.489042804]
	α	0.009091301	0.002221628	[0.009059748; 0.009122855]
	β	2.454931581	0.286086581	[2.450868322; 2.458994840]
Nelder–Mead	a	1.27972e−10	–	–
	b	13.08793619	4.6261741	[13.0808279; 13.0950445]
	α	0.69883693	0.2502418	[0.6884190; 0.7092548]
	β	5.052449497	0.3667520	[4.9210393; 5.1838597]

Table 5 MLE estimates for the parameters of $\mathcal{EW}$ distribution with Nelore data via `BFGS`, `SANN`, and `Nelder-Mead` algorithms

Methods	Parameters	Estimates	SE	Confidence Interval (0.95)
BFGS	b	40.34715566	1.479593e+01	[40.13701054; 40.55730079]
	c	0.01018772	5.397052e−04	[0.01018006; 0.01019539]
	β	2.85355684	1.892528e−01	[2.85086890; 2.85624478]
SANN	b	4.109734969	0.8629722253	[4.097478262; 4.121991676]
	c	0.007648857	0.0003420323	[0.007643999; 0.007653715]
	β	2.942210704	0.2948368954	[2.938023166; 2.946398243]
Nelder–Mead	b	58.782451666	67.03485145	[56.87827328; 60.68663006]
	c	0.009599516	0.00169215	[0.00955145; 0.00964758]
	β	3.443419884	0.81707386	[3.42021025; 3.46662952]

Algorithm 1: Broyden–Fletcher–Goldfarb–Shanno (`BFGS`)

Input: $\boldsymbol{\theta}_0$, f and C
▷$\boldsymbol{\theta}_0$ is the initial solution, f is the evaluation function, C contains control parameters ($fnscale = -1$, $maxiter = 100$, $\eta = 0.001$, $\varepsilon = 0.00001$)

$fnscale \leftarrow getRescale(C)$ ▷Maximize f: $fnscale \leftarrow -1$
$maxit \leftarrow getMaxit(C)$ ▷Maximum number of iterations
$\eta \leftarrow getStepSize(C)$ ▷$\eta > 0$: step size parameter
$\varepsilon \leftarrow getTolerance(C)$ ▷$\varepsilon > 0$: adjust the method's tolerance
$i \leftarrow 0$ ▷i: counter of iterations
$\boldsymbol{g}_i \leftarrow \nabla f(\boldsymbol{\theta}_i)$ ▷Gradient of f at the point $\boldsymbol{\theta}_i$
$H_i \leftarrow I_{p\times p}$ ▷$I_{p\times p}$: identity matrix
while $\left(i < maxit\right) \vee \left(\left(\|\boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i\|/\|\boldsymbol{\theta}_i\| > \varepsilon\right) \wedge \left(\|\boldsymbol{g}_i\| > \varepsilon\right)\right)$ **do**
  $\boldsymbol{d}_i \leftarrow -H_i\boldsymbol{g}_i$ ▷Search direction
  $\alpha_i \leftarrow \arg\min_{\eta>0} f(\boldsymbol{\theta}_i + \eta\boldsymbol{d}_i)$ ▷Wolf linear search
  $\boldsymbol{\theta}_{i+1} \leftarrow \boldsymbol{\theta}_i + \alpha_i\boldsymbol{d}_i$ ▷Update the solution
  $\boldsymbol{g}_{i+1} \leftarrow \nabla f(\boldsymbol{\theta}_{i+1})$ ▷Gradient of f in $\boldsymbol{\theta}_{i+1}$
  $\boldsymbol{\delta}_i \leftarrow \boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i = \alpha_i\boldsymbol{d}_i$ ▷$\boldsymbol{\delta}_i = \left[\nabla^2 f(\boldsymbol{x}_{i+1})\right]^{-1}\boldsymbol{\gamma}_i$
  $\boldsymbol{\gamma}_i \leftarrow \boldsymbol{g}_{i+1} - \boldsymbol{g}_i$ ▷Compute the gradient change
  $H_{i+1} \leftarrow H_i + \left(1 + \frac{\boldsymbol{\gamma}_i^\top H_i\boldsymbol{\gamma}_i}{\boldsymbol{\delta}_i^\top\boldsymbol{\gamma}_i}\right)\frac{\boldsymbol{\delta}_i\boldsymbol{\delta}_i^\top}{\boldsymbol{\delta}_i^\top\boldsymbol{\gamma}_i} - \frac{\boldsymbol{\delta}_i\boldsymbol{\gamma}_i^\top H_i + H_i\boldsymbol{\gamma}_i\boldsymbol{\delta}_i^\top}{\boldsymbol{\gamma}_i^\top\boldsymbol{\delta}_i}$ ▷Update the Hessian
  $i \leftarrow i + 1$ ▷Update the iteration
end
return $\boldsymbol{\theta}_i$ ▷The best solution

SANN algorithm

Annealing is the physical process used to melt metals, which are heated to high temperatures and then cooled slowly, producing a homogeneous material. The simulated annealing (SA) algorithm was originally proposed by [30], being developed later by [31] in the context of optimization problem. The SANN is a variant of SA given in [32], and its pseudo-code is presented in Algorithm 2, adapted from [33].

Algorithm 2: Simulated Annealing Algorithm (SANN)

Input: $\boldsymbol{\theta}_0$, f and C
▷$\boldsymbol{\theta}_0$ is the initial solution, f is the evaluation function, C contains control parameters ($fnscale = -1$, $maxiter = 10000$, $T = 10$, $tmax = 10$)
$fnscale \leftarrow getRescale(C)$ ▷Maximize f: $fnscale \leftarrow -1$
$maxit \leftarrow getMaxit(C)$ ▷Maximum number of iterations
$T \leftarrow getStartTemp(C)$ ▷Initial temperature
$tmax \leftarrow getNumberEval(C)$ ▷Number of function evaluations at each temperature
$i \leftarrow 0$ ▷i: counter of iterations
$f \leftarrow f(\boldsymbol{\theta}_i)$ ▷Evaluation of $\boldsymbol{\theta}_i$
while $i < maxit$ **do**
  for $j = 1 \rightarrow tmax$ **do**
    ▷Generate a solution $\boldsymbol{\theta}'$ in the neighborhood of $\boldsymbol{\theta}_i$ (might depend on T)
    $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}_i + \triangle\boldsymbol{\theta}_i$
    $f' \leftarrow f(\boldsymbol{\theta}')$ ▷Evaluation of $\boldsymbol{\theta}'$
    $r \leftarrow \mathcal{U}(0,1)$ ▷Random number, Uniform(0,1)
    $p \leftarrow \exp\left(\frac{f'-f}{T}\right)$ ▷$P(\boldsymbol{\theta}_i, \boldsymbol{\theta}', T)$ (Metropolis function)
    if $f(\boldsymbol{\theta}') < f(\boldsymbol{\theta}_i) \vee r < p$ **then**
      $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}'$ ▷Update the solution
    end
    if $f(\boldsymbol{\theta}') < f(\boldsymbol{\theta}_i)$ **then**
      $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}'$ ▷Update the solution
    end
    $i \leftarrow i + 1$ ▷Update the iteration
  end
  $T \leftarrow \frac{T}{log(i/tmax) \times tmax + \exp(1)}$ ▷Decrease the temperature
end
return $\boldsymbol{\theta}_i$ ▷The best solution

Nelder–Mead algorithm

The [34] simplex method is an algorithm of unconstrained optimization that belongs to a more general class of direct search whose objective is to find the minimum of a function *f*. Algorithm 3 shows the pseudo-code of Nelder–Mead algorithm [35].

Algorithm 3: `Nelder-Mead` Algorithm

Input: $\boldsymbol{\theta}_0$, f and C
▷$\boldsymbol{\theta}_0$ is the initial solution, f is the evaluation function, C contains control parameters ($fnscale = -1$, $\alpha = 1$, $\beta = 1/2$, $\gamma = 2$, $\delta = 1/2$, $maxit = 500$)
▷$\alpha > 0$: reflection coefficient; $0 < \beta < 1$: contraction coefficient; $\gamma > 1$: expansion coefficient; $0 < \delta < 1$: reduction coefficient

$fnscale \leftarrow getRescale(C)$ ▷Maximize f: $fnscale \leftarrow -1$
$maxit \leftarrow getMaxit(C)$ ▷Maximum number of iterations
$i \leftarrow 0$ ▷i: Counter of iterations
while $i < maxit$ **do**
  Determine $\boldsymbol{\theta}_h$, $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}_l$, $\boldsymbol{\theta}_m$, $f_h \leftarrow f(\boldsymbol{\theta}_h)$, $f_s \leftarrow f(\boldsymbol{\theta}_s)$ and $f_l \leftarrow f(\boldsymbol{\theta}_l)$
  $\boldsymbol{\theta}_r \leftarrow \boldsymbol{\theta}_m + \alpha(\boldsymbol{\theta}_m - \boldsymbol{\theta}_h)$ ▷Reflection: $f_r \leftarrow f(\boldsymbol{\theta}_r)$
  if $f_r < f_l$ **then**
    $\boldsymbol{\theta}_e \leftarrow \boldsymbol{\theta}_m + \gamma(\boldsymbol{\theta}_r - \boldsymbol{\theta}_m)$ ▷Expansion: $f_e \leftarrow f(\boldsymbol{\theta}_e)$
    if $f_e < f_r$ **then**
      $\boldsymbol{\theta}_h \leftarrow \boldsymbol{\theta}_e$
    else
      $\boldsymbol{\theta}_h \leftarrow \boldsymbol{\theta}_r$
    end
  else
    if $f_r \leq f_s$ **then**
      $\boldsymbol{\theta}_h \leftarrow \boldsymbol{\theta}_r$
    else
      if $f_r < f_h$ **then**
        $\boldsymbol{\theta}_h \leftarrow \boldsymbol{\theta}_r$
      end
      $\boldsymbol{\theta}_c \leftarrow \boldsymbol{\theta}_m + \beta(\boldsymbol{\theta}_h - \boldsymbol{\theta}_m)$ ▷Contraction: $f_c \leftarrow f(\boldsymbol{\theta}_c)$
      if $f_c > f_h$ **then**
        $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_l + \delta(\boldsymbol{\theta}_i - \boldsymbol{\theta}_l)$ ▷Reduction: $\forall\ i = 1, \cdots, p+1$
      else
        $\boldsymbol{\theta}_h \leftarrow \boldsymbol{\theta}_c$
      end
    end
  end
  $i \leftarrow i + 1$ ▷Update the iteration
end
return $\boldsymbol{\theta}_i$ ▷The best solution

fitDist function

Algorithm 4 shows the pseudo-code to the *fitDist* function. This function is used to obtain parameter estimates as well as their log-likelihood, variance, confidence interval and MSE.

Algorithm 4: *fitDist* function

```
Function fitDist(x, dist, method, start, par, ntries, hessian):
    ▷x: is the dataset; dist: is the ditribution used (ℰ𝒲, ℰ𝒢𝒲);
     method: is the method used (BFGS, SANN, Nelder-Mead); start: is
     the vector of initial values; par: is the vector of original
     parameters; ntries: is the number of trials to overcome restrictions;
     hessian: logical (TRUE or FALSE). The Hessian matrix must be
     returned? The default is TRUE.
    fit ← Null
    H ← Null
    convergence ← Null
    k ← 0                                                  ▷k: counter of tries
    while k < ntries do
        fit ← optim(start, loglik, x, control, method, hessian)      ▷R function
        par ← getPar(fit)                           ▷The best set of parameters
        loglik ← getLoglik(fit)           ▷The log-likelihood at the point par
        H ← getHessian(fit)                               ▷The Hessian matrix
        var ← getVar(H)                ▷The variance of par: diag(solve(−H))
        ci ← getCI(par, var)                  ▷The confidence interval for par
        sqm ← getMSE(par)                                     ▷The MSE for par
         ▷Get the convergence code: 0 indicates successfull and 1
          indicates that maxit was reached
        convergence ← getConvergence(fit)
         ▷Conditional statement
        if isNull(fit) ∨ isNull(H) ∨ convergence ≠ 0 then
            k ← k + 1                                       ▷Update the tries
        end
    end
return list(par, loglik, var, ci, mse, convergence)          ▷The best solution
```

getSimulation function

Algorithm 5 shows the pseudo-code to the function *getSimulation*. This is the main routine for generating the simulations of the distributions.

Algorithm 5: *getSimulation* function

```
    ▷n: sample size; dist: is the distribution used (ℰ𝒲, ℰ𝒢𝒲); start: is
     the vector of initial values; par: is the vector of original
     parameters; nrep: is the number of replications; ntries is the number
     of trials to overcome restrictions; hessian: logical (TRUE or FALSE).
     The Hessian matrix must be returned? The default is TRUE
Function getSimulation(n, dist, start, par, nrep, ntries, hessian):
    j ← 0                                        ▷j: counter of replications
    while j < nrep do
        x ← random(n, par, dist)                   ▷Sample of size n from dist
        f1[j]←fitDist(x, dist, method=BFGS, start, par, ntries, hessian)
        f2[j]←fitDist(x, dist, method=SANN, start, par, ntries, hessian)
        f3[j]←fitDist(x, dist, method=N.-M., start, par, ntries, hessian)

        nullfit ← any(isNull(f1 [j]) ∨ isNull(f2 [j]) ∨ isNull(f3 [j]))
        if isFalse(nullfit) then
            j ← j + 1                                ▷Update the replications
        end
    end
return list (⟨par⟩, ⟨var⟩, ⟨mse⟩)     ▷The best solution in nrep replications
```

Abbreviations
cdf: cumulative distribution function; hrf: hazard rate function; $\mathcal{EW}$:: exponentiated Weibull; $\mathcal{EGW}$:: exponentiated generalized Weibull; qf: quantile function; pdf: probability density function; BFGS: Brogden–Fletcher–Golfarb–Shanno; SANN: simulated annealing; SE: standard error; MSE: mean squared error.

Acknowledgements
Not applicable.

Authors' contributions
FRSG (was a major contributor in writing the manuscript) involved in writing—original draft, methodology and investigation; FGS involved in writing—review and editing, supervision and validation; CCRB involved in visualization, supervision and validation; FVJS involved in methodology and visualization; JSJ involved in methodology and software; SFAXJ involved in writing—review and editing and visualization; PRDM involved in methodology and software. All authors read and approved the final manuscript.

Funding
No funding was received.

Availability of data and materials
All data generated or analyzed during this study are included in this published article.

Declarations

Competing interests
The authors declare that they have no competing interests

Author details
[1]DEINFO, Universidade Federal Rural de Pernambuco, Recife, Brazil. [2]Instituto Federal de Pernambuco, Campus Recife, Recife, Brazil. [3]Departamento de Estatística, Universidade Estadual da Paraíba, Campina Grande, Brazil. [4]Departamento de Estatística, Universidade Federal da Paraíba, João Pessoa, Brazil.

Received: 16 October 2020 Accepted: 28 September 2021
Published online: 23 October 2021

References
1. Mudholkar, G.S., Srivastava, D.K., Freimer, M.: The exponentiated Weibull family: a reanalysis of the bus-motor-failure data. Technometrics **37**(4), 436–445 (1995)
2. Xie, M., Lai, C.D.: Reliability analysis using an additive Weibull model with bathtub-shaped failure rate function. Reliab. Eng. Syst. Saf. **52**(1), 87–93 (1996)
3. Carrasco, J.F., Ortega, E.M.M., Cordeiro, G.M.: A generalized modified Weibull distribution for lifetime modeling. Comput. Stat. Data Anal. **53**(2), 450–462 (2008)
4. Gusmão, F.R.S., Ortega, E.M.M., Cordeiro, G.M.: The generalized inverse Weibull distribution. Stat. Pap. **52**, 591–619 (2011)
5. Cordeiro, G.M., Ortega, E.M.M., Cunha, D.C.: The exponentiated generalized class of distributions. J. Data Sci. **11**, 1–27 (2013)
6. Eugene, N., Lee, C., Famoye, F.: Beta-normal distribution and its applications. Commun. Stat. Theory Methods **31**, 497–512 (2002)
7. Cordeiro, G.M., Lemonte, A.J.: The exponentiated generalized Birnbaum–Saunders distribution. Appl. Math. Comput. **247**, 762–779 (2014)
8. Silva, R., Gomes-Silva, F., Ramos, M., Cordeiro, G.M.: A new extended gamma generalized model. Int. J. Pure Appl. Math. **100**(2), 309–335 (2015)
9. Andrade, T., Rodrigues, H., Bourguignon, M., Cordeiro, G.M.: The exponentiated generalized Gumbel distribution. Rev. Colomb. Estad. **38**(1), 123–143 (2015)
10. Gomes-Silva, F., Silva, R., Percontini, A., Ramos, M., Cordeiro, G.M.: An extended Dagum distribution: properties and applications. Int. J. Appl. Math. Stat. **56**, 35–56 (2017)
11. Oguntunde, P.E., Odetunmibi, O.A., Adejumo, A.O.: On the exponentiated generalized Weibull distribution: a generalization of the Weibull distribution. Indian J. Sci. Technol. **8**(35), 67611 (2015)
12. Andrade, T.A.N., Bourguignon, M., Cordeiro, G.M.: The exponentiated generalized extended exponential distribution. J. Data Sci. **14**, 393–414 (2016)
13. Cordeiro, G.M., Lemonte, A., Campelo, A.K.: Extended arcsine distribution to proportional data: properties and applications. Stud. Sci. Math. Hung. **53**, 440–466 (2017)
14. Cordeiro, G.M., Andrade, T.A.N., Bourguignon, M., Gomes-Silva, F.: The exponentiated generalized standardized half-logistic distribution. Int. J. Stat. Probab. **6**, 24–42 (2017)
15. Andrade, T., Zea, L.: The exponentiated generalized extended pareto distribution. J. Data Sci. **16**, 781–800 (2018)
16. Andrade, T., Gomes-Silva, F., Zea, L.: Mathematical properties, application and simulation for the exponentiated generalized standardized Gumbel distribution. Acta Sci. Technol. 41, 1807–8664 (2019)
17. Andrade, T., Chakraborty, S., Handique, L., Gomes-Silva, F.: The exponentiated generalized extended Gompertz distribution. J. Data Sci. **17**, 299–330 (2019)
18. Casella, G., Berger, R.L.: Statistical Inference. Duxbury Press, Belmont (2001)

19. Jones, G., Johnson, W.O., Hanson, T.E., Christensen, R.: Identifiability of models for multiple diagnostic testing in the absence of a gold standard. Biometrics **66**(3), 855–863 (2010)
20. Paulino, C.D., Pereira, C.A.B.: On identifiability of parametric statistical models. Stat. Methods Appl. J. Ital. Stat. Soc. **3**, 125–151 (1994)
21. Mudholkar, G.S., Srivastava, D.K.: Exponentiated Weibull family for analyzing bathtub failure-rate data. IEEE Trans. Reliab. **42**(2), 299–302 (1993)
22. Jiang, R., Murthy, D.N.P.: The exponentiated Weibull family: a graphical approach. IEEE Trans. Reliab. **48**(1), 68–72 (1999)
23. Gusmão, F.R.S., Ortega, E.M.M., Cordeiro, G.M.: Reply to the Letter to the Editor of M. C. Jones. Stat. Pap. **53**, 252–254 (2012)
24. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (2012)
25. Colosimo, E.A., Giolo, S.R.: Análise de sobrevivência aplicada. Edgar Blucher, São Paulo (2006)
26. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms 1. General considerations. IMA J. Appl. Math. **6**(1), 76–90 (1970)
27. Fletcher, R.: A new approach to variable metric algorithms. Comput. J. **13**(3), 317–322 (1970)
28. Goldfarb, D.: A family of variable metric updates derived by variational means. Math. Comput. **24**(109), 23–26 (1970)
29. Shanno, D.F.: Conditioning of quasi-Newton methods for function minimization. Math. Comput. **24**(111), 647–656 (1970)
30. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
31. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
32. Bélisle, C.J.P.: Convergence theorems for a class of simulated annealing algorithms on $\mathcal{R}^d$. J. Appl. Probab. **29**(4), 885–895 (1992)
33. Cortez, P.: Modern Optimization with R. Springer, Berlin (2014)
34. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
35. Givens, G.H., Hoeting, J.A.: Computational Statistics, 2nd edn. Wiley, London (2013)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.