Anais da Academia Brasileira de Ciências (2019) 91(1): e20180480
(Annals of the Brazilian Academy of Sciences)
Printed version ISSN 0001-3765 / Online version ISSN 1678-2690
http://dx.doi.org/10.1590/0001-3765201920180480
www.scielo.br/aabc | www.fb.com/aabcjournal

A new class of lifetime models and the evaluation of the confidence intervals by double percentile bootstrap

PEDRO R.D. MARINHO[1], MARCELO BOURGUIGNON[2], RODRIGO B. SILVA[1] and GAUSS M. CORDEIRO[3]

[1]Departamento de Estatística, Universidade Federal da Paraíba, Cidade Universitária, s/n, 58051-900 João Pessoa, PB, Brazil
[2]Departamento de Estatística, Universidade Federal do Rio Grande do Norte, Lagoa Nova, 59078-970 Natal, RN, Brazil
[3]Departamento de Estatística, Universidade Federal de Pernambuco, s/n, Cidade Universitária, 50740-540 Recife, PE, Brazil

Manuscript received on May 15, 2018; accepted for publication on July 30, 2018

How to cite: MARINHO PRD, BOURGUIGNON M, SILVA RB AND CORDEIRO GM. 2018. A new class of lifetime models and the evaluation of the confidence intervals by double percentile bootstrap. An Acad Bras Cienc 91: e20180480. DOI 10.1590/0001-3765201920180480.

Abstract: In this paper, we introduce a new three-parameter distribution by compounding the Nadarajah-Haghighi and geometric distributions, which can be interpreted as a truncated Marshall-Olkin extended Weibull. The compounding procedure is based on the work by Marshall and Olkin 1997. We prove that the new distribution can be obtained as a compound model with mixing exponential distribution. It can have decreasing, increasing, upside-down bathtub, bathtub-shaped, constant and decreasing-increasing-decreasing failure rate functions depending on the values of the parameters. Some mathematical properties of the new distribution are studied including moments and quantile function. The maximum likelihood estimation procedure is discussed and a particle swarm optimization algorithm is provided for estimating the model parameters. The flexibility of the new model is illustrated with an application to a real data set.

Key words: Exponential distribution, Failure rate function, Geometric distribution, Maximum likelihood estimation, Nadarajah-Haghighi distribution.

INTRODUCTION

Nadarajah and Haghighi (2011) introduced and studied the mathematical properties of an extension of the exponential distribution that allows for increasing, decreasing and constant hazard rate functions (hrfs). The model is referred to as the Nadarajah-Haghighi (NH) distribution. Its cumulative distribution function (cdf) is given by

$$G(t;\alpha,\lambda) = 1 - \exp\{1 - (1+\lambda\, t)^{\alpha}\}, \quad t > 0, \tag{1}$$

Correspondence to: Marcelo Bourguignon
E-mail: m.p.bourguignon@gmail.com
ORCid: 0000-0002-1182-5193

where $\lambda > 0$ is the scale parameter and $\alpha > 0$ is the shape parameter. If T follows the Nadarajah-Haghighi distribution, we shall denote by $T \sim \text{NH}(\alpha, \lambda)$. The corresponding probability density function (pdf) is given by

$$g(t; \alpha, \lambda) = \alpha\,\lambda\,(1 + \lambda\,t)^{\alpha-1} \exp\{1 - (1 + \lambda\,t)^{\alpha}\}, \quad t > 0. \tag{2}$$

The NH distribution presents several advantages if compared with some well-known generalizations of the exponential model, such as the gamma, Weibull and exponentiated exponential (EE) distributions. For example, unlike these distributions, the NH distribution allows for an increasing hrf when its corresponding density is monotonically decreasing. The second advantage is the ability to model data that have their mode fixed at zero. Another advantage is based on a mathematical relationship with the Weibull distribution, where the NH model can be interpreted as a truncated Weibull distribution. These three facts combined may attract more complex applications in the literature of lifetime distributions.

The NH distribution is not the only extension of the exponential distribution. In addition to the gamma and Weibull distributions, many other generalizations have been proposed over the years. One of them is

$$G(t; \rho, \lambda, \alpha) = (1 - \rho\,\mathrm{e}^{-\lambda\,t})^{\alpha}, \quad t > \lambda^{-1} \log \rho, \tag{3}$$

where $\rho > 0$, $\lambda > 0$ and $\alpha > 0$. Note that the EE distribution, discussed in Gupta et al. (1998), is actually a particular case of equation (3) when $\rho = 1$. Nadarajah and Kotz (2006) and Barreto-Souza et al. (2010) proposed two important extensions of the exponential model, which are the beta exponential (BE) and beta generalized exponential (BGE) distributions, respectively. The BE and BGE distributions are special cases of the beta family of distributions, pionnered by Eugene et al. (2002). The beta family still contains other generalizations of the exponential model such as the beta Weibull (BW) (Lee et al. 2007) and beta modified Weibull (BMW) (Silva et al. 2010) distributions. The Kumaraswamy class, introduced by Cordeiro and de Castro (2011), also contains several models that extend the exponential distribution, such as the Kumaraswamy Weibull (KwW) (Cordeiro et al. 2010) and Kumaraswamy generalized Rayleigh (KwGR) (Gomes et al. 2014) distributions. References about other generalizations of the exponential model are widespread and the reader can see those listed in the above papers.

In this paper, we introduce a new continuous distribution, which is an extension of the NH model, by compounding the NH and geometric distributions. The new model is therefore another extension of the exponential distribution and is referred to as the Nadarajah-Haghighi geometric (NHG) distribution. The proposed distribution is more flexible for modeling lifetime data, namely in reliability, in terms of its failure rate shapes, which maybe be constant, decreasing, increasing, upside-down bathtub and bathtub shaped. The compounding procedure follows the pioneering work by Marshall and Olkin (1997a). In the same way, several classes of distributions were proposed by compounding some useful lifetime and power series (PS) distributions in the last few years. Chahkandi and Ganjali (2009) introduced the exponential power series (EPS) class of distributions, which contains as special cases the exponential Poisson (EP), exponential geometric (EG) and exponential logarithmic (EL) distributions. Morais and Barreto-Souza (2011) defined the Weibull power series (WPS) class, which includes, as sub-models, the EPS distributions. The WPS distributions can have increasing, decreasing and upside down bathtub hrfs. The generalized exponential power series (GEPS) distributions were proposed by Mahmoudi and Jafari (2012) following the same approach of Morais and Barreto-Souza (2011). Silva et al. (2013) studied the extended Weibull power series (EWPS) family, which includes as special models the EPS and WPS families. Bourguignon et al. (2014)

extended the Birnbaum-Saunders distribution through the class of Birnbaum-Saunders power series (BSPS) distributions. In a recent paper, (Silva and Cordeiro 2015) introduced the Burr XII power series (BXIIPS) family of distributions.

The rest of the paper is organized as follows. Section 2 introduces the NHG distribution and presents some special models. Further, we demonstrate that the NHG density function can be expressed as a mixture of densities of minimum of the baseline order statistics and study some of its properties. In Section 3, the estimation procedure is approached by maximum likelihood via a particle swarm optimization approach. Section 4 deals with bootstrap percentile intervals. Numerical results from Monte Carlo simulation experiments are investigated in Section 5. We also perform simulations to assess the use of bootstrap percentiles (simple and double) for construction of confidence intervals for the parameters. An application to real data is performed in Section 6. Finally, concluding remarks are presented in Section 7.

CONSTRUCTION OF THE NHG DISTRIBUTION

Let $T_1, \ldots, T_N$ be independent and identically distributed (iid) NH random variables with cdf (1) and pdf (2). We assume that N has a zero-truncated geometric distribution independent of the T's with probability mass function (pmf) given by

$$p_n = P(N = n) = (1 - p)\, p^{n-1},\ n = 1, 2, \ldots,$$

with $p \in (0, 1)$. Defining $X = \min(T_1, \ldots, T_N)$, the conditional random variable $(X|N = n)$ has cdf

$$P(X \leq x, N = n) = (1 - p)\, p^{n-1}(1 - \{\exp[1 - (1 + \lambda\, x)^{\alpha}]\}^n), \quad x > 0, \quad n \geq 1.$$

The NHG distribution is defined by the marginal cdf of X,

$$F(x; \alpha, \lambda, p) = \frac{1 - \exp[1 - (1 + \lambda\, x)^{\alpha}]}{1 - p \exp[1 - (1 + \lambda\, x)^{\alpha}]}, \quad x > 0. \tag{4}$$

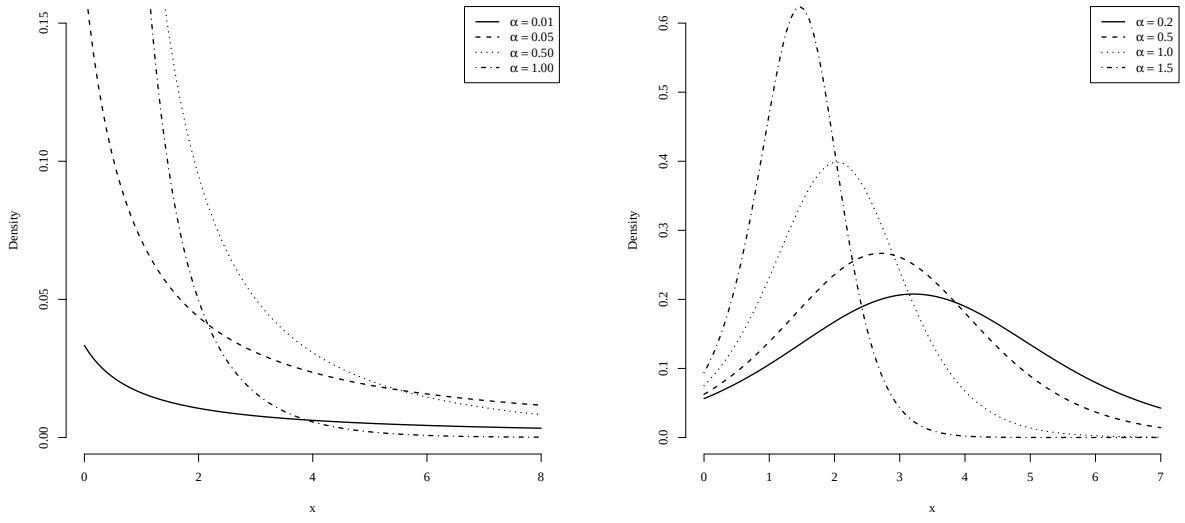
Hereafter, the random variable X according (4) with parameters α, λ and p is denoted by $X \sim \text{NHG}(\alpha, \lambda, p)$. The pdf of X is

$$f(x; \alpha, \lambda, p) = \frac{(1 - p)\, \alpha\, \lambda\, (1 + \lambda\, x)^{\alpha - 1} \exp[1 - (1 + \lambda\, x)^{\alpha}]}{\{1 - p \exp[1 - (1 + \lambda\, x)^{\alpha}]\}^2}, \quad x > 0. \tag{5}$$

It can be shown that

$$\lim_{x \to 0} f(x; \alpha, \lambda, p) = \alpha\, \lambda / (1 - p) \ \text{ and } \ \lim_{x \to \infty} f(x; \alpha, \lambda, p) = 0.$$

Even when $p \leq 0$, Equation (5) is a density function. We can then define the NHG distribution by (5) for any $p < 1$. The study of the new distribution is important since it extends some distributions previously considered in the literature. In fact, the NH distribution is obtained by taking $p = 0$, please see (Nadarajah and Haghighi 2011). The EG distribution (Adamidis and Loukas 1998) follows by taking $\alpha = 1$ and $0 < p < 1$, whereas the EEG distribution (Adamidis et al. 2005). is obtained when $\alpha = 1$ for any $p < 1$. Clearly, the EEG distribution extends the EG distribution. For $\alpha = 1$ and $p = 0$, Equation (5) reduces to the exponential distribution. When $p \to 1^-$ (p tending to 1 from by the left), the NHG distribution distribution converges to a distribution degenerated at zero, i.e., $P(X = 0) = 1$. Hence, the parameter p can be interpreted as a degeneration parameter. Figure 1 displays the pdf of X for selected parameter values.



(a) $p = -15$ **(b)** $p = 0.9$

Figure 1 - The NHG density function for some parameter values; $\lambda = 1$.

Proposition 1. *For any $\lambda > 0$, the pdf of the NHG distribution is monotonous if $\alpha > 1$ and $p \leq 0$ and the pdf is strictly decreasing if $\alpha < 1$ and $0 < p < 1$.*

Proof. The proof of this result follows directly by derivation criteria. □

Proposition 2. *The NHG density function is log-convex if $\alpha < 1$ and $0 \leq p < 1$, and it is log-concave if $\alpha > 1$ and $p \leq 0$.*

Proof. Let $z = (1 + \lambda\, x)^{\alpha}$. It implies that $z > 1$ for $x > 0$. We have $x = (z^{1/\alpha} - 1)/\lambda$. Now, rewriting the NHG density function as a function of z, $\delta(z)$ say, we obtain

$$\delta(z) = \frac{(1-p)\,\alpha\,\lambda\, z^{(\alpha-1)/\alpha}\, \mathrm{e}^{1-z}}{[1 - p\,\mathrm{e}^{1-z}]^2}, \quad z > 1.$$

As a simple derivation exercise, deriving with respect to z, it is possible to observe that $\log[\delta(z)] < 0$.

The second derivative of $\log[\delta(z)]$ with respect to z is given by

$$\frac{d^2\, \log[\delta(z)]}{d\, z^2} = -\left[\frac{(\alpha-1)}{\alpha\, z^2} - \frac{2\, p\, \mathrm{e}^{1-z}}{[1 - p\,\mathrm{e}^{1-z}]^2}\right].$$

□

The main motivation for the new distribution is based on four points:

1. Ability (or the inability) of the NHG distribution to model data that have their mode fixed at zero.
2. As we shall see later, the NHG hrf can be constant, decreasing, increasing, decreasing-increasing-decreasing, upside-down bathtub, bathtub-shaped or constant.

3. If Y is a Marshall-Olkin extended Weibull random variable with shape parameter α and scale parameter λ, then the density in (5) is the same as that one of the random variable $Z = Y - \lambda^{-1}$ truncated at zero; that is, the NHG distribution can be interpreted as a truncated Marshall-Olkin extended Weibull distribution.

4. It can be applied in some interesting situations as follows:

   • Time to relapse of cancer under the first-activation scheme. Here, N is the number of carcinogenic cells for an individual left active after the initial treatment and X_i is the time spent for the ith carcinogenic cell to produce a detectable cancer mass, for $i \geq 1$;

   • Time to the first failure. Suppose that the failure of a device occurs due to the presence of an unknown number N of initial defects of same kind, which can be identifiable only after causing failure and are repaired perfectly;

   • Reliability. From the stochastic representations $X = \min\{X_i\}_{i=1}^N$ and $Z = \max\{X_i\}_{i=1}^N$, we note that the NHG model can arise in series and parallel systems with identical components, which appear in many industrial applications and biological organisms.

Proposition 3. *The distribution of the form (4) is geometric extreme stable.*

Proof. The proof follows easily using the arguments by Marshall and Olkin (1997b). So, we omitted the details. □

Proposition 4. *The density function of X can be expressed as a mixture of densities of minimum order statistics of T.*

Proof. For any positive real number a, and for $|z| < 1$, we have the generalized binomial expansion (Prudnikov et al. 1986).

$$(1-z)^{-a} = \sum_{n=0}^{\infty} \frac{\Gamma(a+n)}{\Gamma(a)\, n!} z^n, \tag{6}$$

where $\Gamma(\cdot)$ is the gamma function. Applying (6) to (5), yields

$$f(x;\alpha,\lambda,p) = \sum_{n=1}^{\infty} p_n\, f_{T_{(1)}}(x;\alpha,\lambda,n), \quad x > 0,$$

where $\sum_{n=1}^{\infty} p_n = 1$ and $f_{T_{(1)}}(t;\alpha,\lambda,n)$ is the density function of $T_{(1)} = \min(T_1,\ldots,T_n)$, for fixed n, given by

$$f_{T_{(1)}}(t;\alpha,\lambda,n) = n\,\alpha\,\lambda\,(1+\lambda\,t)^{\alpha-1}\{\exp[1-(1+\lambda\,t)^{\alpha}]\}^n, \quad t > 0.$$

□

Remark 1. *The cdf and pdf of $Y = \max(T_1,\ldots,T_N)$ are given by*

$$F_Y(y;\alpha,\lambda,p) = \frac{(1-p)\{1-\exp[1-(1+\lambda\,x)^{\alpha}]\}}{1-p\{1-\exp[1-(1+\lambda\,x)^{\alpha}]\}}, \quad y > 0 \tag{7}$$

and

$$f_Y(y;\alpha,\lambda,p) = \frac{(1-p)\,\alpha\,\lambda\,(1+\lambda\,x)^{\alpha-1}\{\exp[1-(1+\lambda\,x)^{\alpha}]\}}{(1-p\{1-\exp[1-(1+\lambda\,x)^{\alpha}]\})^2}.$$

The distribution with cdf (7) is called the *complementary Nadarajah-Haghighi-geometric* (CNHG) distribution. It is a suitable model in a complementary risk problem based in the presence of latent risks, which arise in several areas such as public health, actuarial science, biomedical studies, demography and industrial reliability (Basu and Klein 1982). However, in this work, we do not focus on this alternative class of distributions.

Proposition 5. *Let $X \sim NHG(\alpha, \lambda, p)$. Then:*

i) The cdf of the nth order statistic corresponding to the pdf (5) is given by

$$F_n(x) = [F(x;\alpha,\lambda,p)]^n = \left\{\frac{1-\exp[1-(1+\lambda\,x)^{\alpha}]}{1-p\exp[1-(1+\lambda\,x)^{\alpha}]}\right\}^n;$$

ii) The density function of the nth order statistic is given by

$$f_n(x) = \frac{n(1-p)f(x;\alpha,\lambda,p)[F(x;\alpha,\lambda,p)]^{n-1}}{\{1-p[1-F(x;\alpha,\lambda,p)]\}^{n+1}}.$$

Proof. The proof of Proposition 4 is trivial. □

The NHG survival function is given by

$$S(x;\alpha,\lambda,p) = \frac{(1-p)\exp[1-(1+\lambda\,x)^{\alpha}]}{1-p\exp[1-(1+\lambda\,x)^{\alpha}]}, \quad x>0,$$

and the corresponding hrf (for $x>0$) becomes

$$h(x;\alpha,\lambda,p) = \frac{\alpha\,\lambda\,(1+\lambda\,x)^{\alpha-1}}{1-p\,\exp[1-(1+\lambda\,x)^{\alpha}]} = \frac{h_{NH}(x;\alpha,\lambda)}{1-p\,\exp[1-(1+\lambda\,x)^{\alpha}]},$$

where $h_{NH}(x;\alpha,\lambda)$ is the NH hrf.

Note that the ratio $h(x;\alpha,\lambda,p)/h_{NH}(x;\alpha,\lambda)$ is increasing in x for $p<0$ and decreasing in x for $0<p<1$. Further, we have

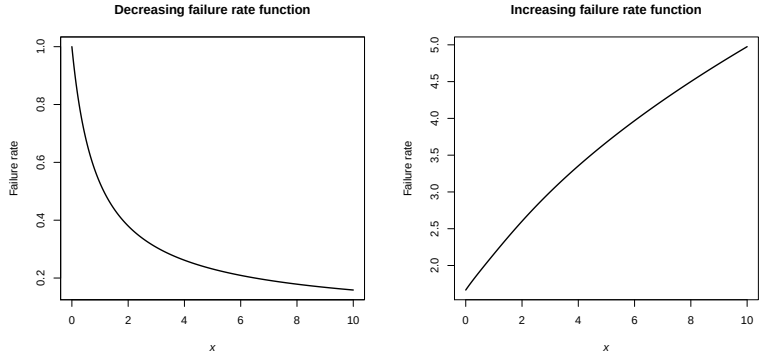
$$\lim_{x\to 0} h(x;\alpha,\lambda,p) = \frac{\alpha\,\lambda}{1-p} \text{ and } \lim_{x\to\infty} h(x;\alpha,\lambda,p) = \begin{cases} 0, & \alpha<1,\\ \infty, & \alpha>1,\\ \alpha\,\lambda, & \alpha=1. \end{cases}$$

Proposition 6. *For $\lambda>0$, the NHG distribution has an increasing hrf if $\alpha>1$ and $p<0$, and it has a decreasing hrf if $\alpha<1$ and $0<p<1$. It is constant if $\alpha=1$ and $p=0$.*

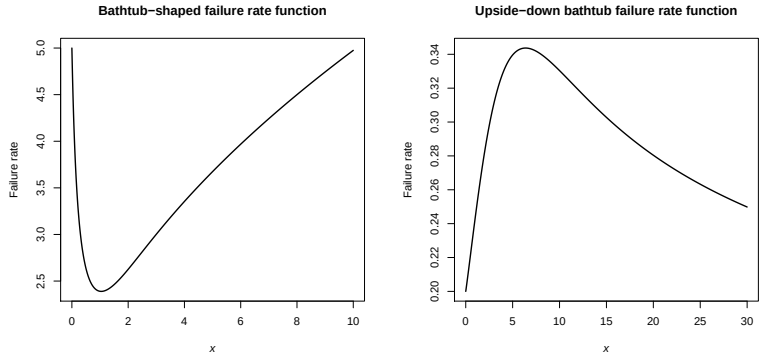
Proof. The proof follows trivially by derivation criteria.

□

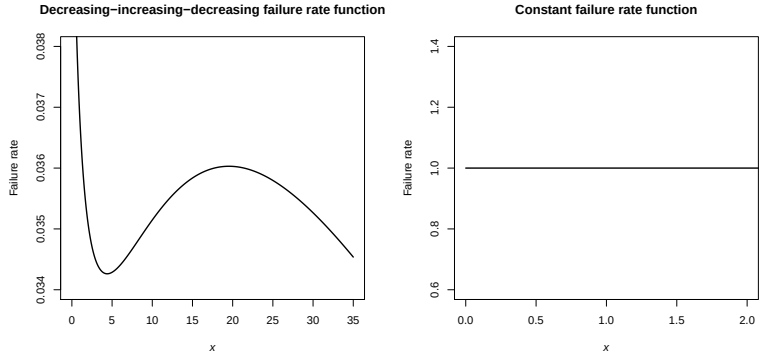
Figure 2 displays some plots of the NHG hrf for some parameter values. The parameter λ does not change the shape of the hrf since it is a scale parameter. It is evident that this hrf can be decreasing, increasing, upside-down bathtub shaped or bathtub-shaped. It is difficult to determine analytically the parameter ranges corresponding to these shapes. It is interesting to point out that the NHG hrf can also be decreasing-increasing-decreasing. So, this distribution is quite flexible and can be used effectively in



(a) $\alpha = 0.5$ and $p = 0.5$ **(b)** $\alpha = 1.5$ and $p = 0.1$



(c) $\alpha = 1.5$ and $p = 0.7$ **(d)** $\alpha = 0.7$ and $p = -2.5$



(e) $\alpha = 0.4$ and $p = -8.5$ **(f)** $\alpha = 1.0$ and $p = 0.0$

Figure 2 - The NHG hrf for some parameter values; $\lambda = 1$.

analyzing real data in several areas. Thus, the beauty and importance of the new distribution lies in its ability to model monotone as well as non-monotone failure rates, which are quite common in reliability and biological studies.

Theorem 7. *Suppose that the conditional survival function of a continuous random variable X is given by*

$$S(x|\theta) = \exp\left\{\theta\left[1 - e^{(1+\lambda x)^{\alpha}-1}\right]\right\}, \quad x > 0,\ \alpha,\ \lambda,\ \theta > 0.$$

If the parameter θ has the exponential distribution with pdf

$$\pi(\theta) = (1-p)e^{-(1-p)\theta}, \quad \theta > 0, \quad p < 1,$$

then the compound distribution of X has the NHG distribution.

Proof.

$$\begin{aligned} S(x;\alpha,\lambda,p) &= \int_0^{\infty} S(x|\theta)\,\pi(\theta)\,\mathrm{d}\theta = (1-p)\int_0^{\infty} \exp\left\{\theta\left[1-\mathrm{e}^{(1+\lambda x)^{\alpha}-1}\right]\right\}\mathrm{e}^{-(1-p)\theta}\,\mathrm{d}\theta \\ &= (1-p)\int_0^{\infty} \exp\left\{-\theta\left[\mathrm{e}^{(1+\lambda x)^{\alpha}-1} - p\right]\right\}\mathrm{d}\theta = \frac{1-p}{\exp[(1+\lambda x)^{\alpha}-1]-p} \\ &= \frac{(1-p)\exp[1-(1+\lambda x)^{\alpha}]}{1-p\exp[1-(1+\lambda x)^{\alpha}]}, \end{aligned}$$

which is the NHG survival function. □

The rth moment of X is given by

$$\mathrm{E}(X^r) = (1-p)\,\mathrm{e}\,I(p,r,\alpha),$$

where $I(p,r,\alpha) = \int_1^{\infty}(u^{1/\alpha}-1)^r \mathrm{e}^{-u}(1-p\,\mathrm{e}^{1-u})^{-2}\mathrm{d}u$ is an integral to be numerically evaluated. This integral can be easily computed numerically in software such as `Julia` and `R`.

The quantile function (qf) corresponding to equation (4) is given by

$$Q(u) = \lambda^{-1}\left\{\left[1 - \log\left(\frac{1-u}{1-u\,p}\right)\right]^{1/\alpha} - 1\right\}, \quad 0 < u < 1.$$

Simulating the NHG random variable is straightforward. Let U be a uniform variate on the unit interval $(0,1)$. Thus, the random variable X is given by

$$X = \lambda^{-1}\left\{\left[1 - \log\left(\frac{1-U}{1-Up}\right)\right]^{1/\alpha} - 1\right\},$$

follows (5), i.e, $X \sim \mathrm{NHG}(\alpha,\lambda,p)$.

MAXIMUM LIKELIHOOD ESTIMATION

In this section, we determine the maximum likelihood estimates (MLEs) of the parameters of the new distribution from complete samples only. Let $x_1,\ldots,x_n$ be observed values from the NHG distribution

with parameters α, λ and p. Let $\boldsymbol{\theta} = (\alpha, \lambda, p)^\top$ be the parameter vector. The total log-likelihood function for $\boldsymbol{\theta}$ is given by

$$\begin{aligned}\ell_n(\boldsymbol{\theta}) &= n + n\,\log[(1-p)\,\alpha\,\lambda] + (\alpha-1)\sum_{i=1}^{n}\log(1+\lambda\,x_i) - \sum_{i=1}^{n}(1+\lambda\,x_i)^{\alpha} \\ &- 2\sum_{i=1}^{n}\log\{1-p\,\exp[1-(1+\lambda\,x_i)^{\alpha}]\}.\end{aligned}$$

By taking the partial derivatives of the log-likelihood function with respect to the parameters in $\boldsymbol{\theta}$, we obtain the components of the score vector $\boldsymbol{U_\theta} = (U_\alpha, U_\lambda, U_p)^\top$:

$$U_\alpha = \frac{n}{\alpha} + \frac{1}{\alpha}\sum_{i=1}^{n}(1-\dot{\nu}_i)\log(\dot{\nu}_i) - \frac{2\,p}{\alpha}\sum_{i=1}^{n}\dot{\pi}_i\,\dot{\nu}_i\log(\dot{\nu}_i),$$

$$U_\lambda = \frac{n}{\lambda} + (\alpha-1)\sum_{i=1}^{n}x_i\,\dot{\nu}_i^{-1/\alpha} - \alpha\sum_{i=1}^{n}x_i\,(2p+\dot{\pi}_i)\,\dot{\nu}_i^{1-1/\alpha} \quad \text{and} \quad U_p = 2\sum_{i=1}^{n}\dot{\pi}_i - \frac{n}{1-p},$$

where

$$\dot{\nu}_i = (1+\lambda\,x_i)^{\alpha} \text{ and } \dot{\pi}_i = \frac{\exp[1-(1+\lambda\,x_i)^{\alpha}]}{1-p\,\exp[1-(1+\lambda\,x_i)^{\alpha}]},\ i=1,\ldots,n.$$

Setting U_α, U_λ and U_p equal to zero and solving the equations simultaneously yields the MLE $\widehat{\boldsymbol{\theta}} = (\widehat{\alpha}, \widehat{\lambda}, \widehat{p})^\top$ of $\boldsymbol{\theta} = (\alpha, \lambda, p)^\top$. These equations cannot be solved analytically and statistical software can be used to solve them numerically using iterative methods such as the Newton-Raphson type algorithms.

The normal approximation for the MLE of $\boldsymbol{\theta}$ can be used for constructing approximate confidence intervals and for testing hypotheses on the parameters α, λ and p. Under conditions that are fulfilled for the parameters in the interior of the parameter space, we can approximate the distribution of $\sqrt{n}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta})$ by the multivariate normal $N_3(\mathbf{0}, \boldsymbol{K}_{\boldsymbol{\theta}}^{-1})$, where $\boldsymbol{K_\theta}$ is the unit expected information matrix. The asymptotic behavior remains valid if $\boldsymbol{K_\theta} = \lim_{n\to\infty} n^{-1}\boldsymbol{J}_n(\boldsymbol{\theta})$, where $\boldsymbol{J}_n(\boldsymbol{\theta})$ is the observed information matrix, is replaced by the average sample information matrix evaluated at $\widehat{\boldsymbol{\theta}}$, i.e., $n^{-1}\boldsymbol{J}_n(\widehat{\boldsymbol{\theta}})$. The elements of the observed information matrix can be obtained from the authors upon request.

OPTIMIZATION ALGORITHM

In computer science, the particle swarm optimization (PSO) is a computational method for optimization of parametric and multiparametric functions.

Further details on the philosophy of the PSO method are given in the book Swarm Intelligence (Kennedy et al. 2001).

The PSO algorithm optimizes a problem by having a population of candidate solutions and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. The movement of the particles in the search space is randomized. For each iteration of the PSO algorithm there is a leader particle, which is the particle that minimizes the objective function in the corresponding iteration. The remaining particles arranged in the search region will follow the leader particle randomly and sweep the area around this leading particle. In this local search process, another particle may become the new leader and the other particles will follow this new leader randomly. Each particle arranged

in the search region has a velocity vector and position vector and its movement in the search region is given by changes in these vectors. The PSO algorithm is presented below, where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function to be minimized, S is the number of particles in the swarm (set of feasible points, i.e. search region), each particle having a vector position $x_i \in \mathbb{R}^n$ in the search-space and a vector velocity defined by $v_i \in \mathbb{R}^n$. Let p_i be the best known position of particle i and g the best position of all particles.

1. For each particle $i = 1, \ldots, S$ do:
   - Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
   - Initialize the particle's best known position to its initial position: $p_i \leftarrowtail x_i$.
   - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrowtail p_i$.
   - Initialize the particle's velocity: $v_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$.
2. Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:
   - For each particle $i = 1, \ldots, S$ do:
     - Pick random numbers: $r_p, r_g \sim U(0, 1)$.
     - For each dimension $d = 1, \ldots, n$ do:
       - Update the particle's velocity: $v_{i,d} \leftarrowtail \omega\, v_{i,d} + \varphi_p r_p(p_{i,d} - x_{i,d}) + \varphi_g r_g(g_d - x_{i,d})$.
     - Update the particle's position: $x_i \leftarrowtail x_i + v_i$
     - If $f(x_i) < f(p_i)$ do:
       - Update the particle's best known position: $p_i \leftarrowtail x_i$
       - If $f(p_i) < f(g)$ update the swarm's best known position: $g \leftarrowtail p_i$.
3. Now g holds the best found solution.

The parameter ω is called inertia coefficient and, as the name implies, controls the inertia of each particle arranged in the search region. The quantities ω_p and ω_g control the acceleration of each particle and are called acceleration coefficients. The PSO algorithm described above, which is implemented in the programming language `R`, is presented below.

This algorithm with few modifications will be implemented in the `AdequacyModel` package, which is available on the website of `R`. The algorithm above is quite general and can be applied to maximize any function involving or not a database. Using `pso`, a given function is maximized taking into consideration vectors of restrictions delimiting the search-space. In truth, the `pso` function above is constructed to minimize any function. However, to maximize f is equivalent to minimize $-f$. A brief description of the arguments of the `pso` function are listed below.

- `func`: Objective function that we want to maximize;
- `S`: Number of particles considered. By default, the number of particles is equal to 150;
- `lim_inf` e `lim_sup`: Vectors that restrict the region-search inferiorly and superiorly, respectively.

- `e`: Error considered. The algorithm stops if the variance in the last 10 iterations is less than or equal to `e`;
- `data`: By default `data=NULL`, but when the `func` is a log-likelihood function, `data` is a data vector;
- `N`: Maximum number of iterations.

One advantage of the PSO method is that we do not need to concern ourselves with initial shots. Problems with initial shots are frequent in methods such as the BFGS when the objective function involves flat or nearly flat regions.

The example below shows clearly this problem and the use of the function `pso`, especially how to specify the objective function for the argument `func`. The PSO method is described in the A.

Example:

Consider Easom function $f(x,y) = -\cos(x)\cos(y)\exp\{-[(x-\pi)^2+(y-\pi)^2]\}$, and $-10 \leq x, y \leq 10$. The Easom function is minimized at $x = y = \pi$, with $f(\pi,\pi) = -1$. The use of the `pso` function to minimize the above function is

```
easom <- function(par,x){
   x1 = par[1]
   x2 = par[2]
   -cos(x1)*cos(x2)*exp(-((x1-pi)^2 + (x2-pi)^2))
}
set.seed(0)
pso(func=easom,S=350,lim_inf=c(-10,-10),lim_sup=c(10,10))
```

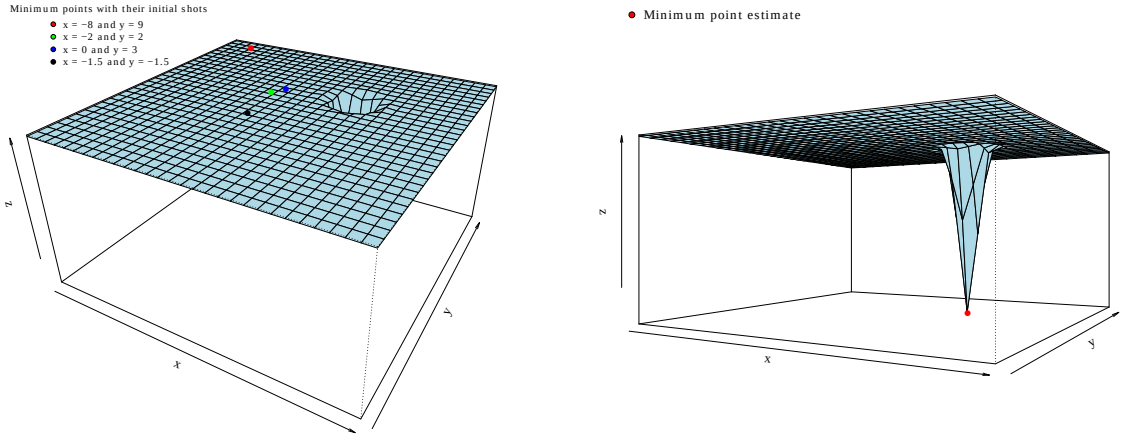
In both minimizations there is convergence. For minimization through BFGS method, we use the `optim` function of the `R` language with the argument `method="BFGS"`. For more details on the `optim` function, do `help(optim)`. Figure 3(a) presents four estimates obtained by the BFGS method, whose points are those of minimum obtained by this method, using the BFGS procedure with different initial shots. In the legend are described the initial shots. In these four estimates for different shots the `optim` function indicates convergence.

Before performing the `pso` function, the seed of the Mersenne-Twister generator proposed by Matsumoto and Nishimura (1998) is fixed at zero, i.e, `set.seed(0)`. Is it possible to perceive clearly through Figure 3 that the minimization method using the PSO function `pso` is much more efficient and approaches better the global minimum.

BOOTSTRAP CONFIDENCE INTERVALS

The bootstrap method was introduced in 1979 by Efron. This method was inspired by a previous methodology based on resampling called jackknife. Efron (1979) summarized the methodologies based on resampling existing until then and established a new research area.

The idea of replacing complicated and often inaccurate simulation methods based on resampling approaches has attracted several researchers to develop methodologies based on bootstrap for various



(a) Minimization by BFGS method **(b)** Minimization by PSO method

Figure 3 - Points of minimum obtained by `optim` and `pso` functions by using BFGS and PSO methods.

purposes. With the popularization of the bootstrap method, some researchers have begun to establish mathematical conditions under which the bootstrap is justifiable.

In the literature there are many jobs that make use of bootstrap methodologies. In general, the bootstrap method is used to correct the biases of the estimators, in construction of confidence intervals, hypothesis tests, estimation of standard errors of estimators, among others.

The bootstrap methods present two approaches, namely the parametric bootstrap and nonparametric bootstrap. Parametric bootstrap refers to the case where the resampling is performed based on a known or established distribution $F(\widehat{\theta})$, where $\widehat{\theta}$ is an estimator of θ. On the other hand, in bootstrap nonparametric, there is the lack of a true distribution F. The resampling is based on the empirical distribution function $\widehat{F}_n$. Resample of $\widehat{F}_n$ is equivalent to resample data with replacement.

BOOTSTRAP PERCENTILE INTERVAL

Let T_n be an estimator of the scale θ based on n observations and t its estimate. Let T_n^* be the same estimator based on n observations resampling from the original sample with replacement and t^* its estimate. For simplicity, suppose T_n is a continuous random variable. Denoting the pth quantile of the distribution of the random variable $T_n - \theta$ by a_p, we obtain

$$\Pr\left\{T_n - \theta \leq a_{\frac{\alpha_1}{2}}\right\} = \Pr\left\{T_n - \theta \geq a_{1-\frac{\alpha_2}{2}}\right\} = \frac{\alpha}{2}.$$

As the amount $Q = T_n - \theta$ is invertible and T_n depends only on the sample, we can construct the confidence interval for θ by rewriting the events above, i.e., we can replace the events $T_n - \theta \leq a_{\frac{\alpha_1}{2}}$ and $T_n - \theta \geq a_{1-\frac{\alpha_2}{2}}$ by $\theta > T_n - a_{\frac{\alpha_1}{2}}$ and $\theta < T_n - a_{1-\frac{\alpha_2}{2}}$, respectively. Thus, the confidence interval of level γ is given by the limits

$$\ell_{\alpha/2} = t - a_{1-\frac{\alpha_2}{2}},\ \ell_{1-\alpha/2} = t - a_{\frac{\alpha_1}{2}}.$$

In many situations, we do not know the distribution of $T_n - \theta$. In such cases, suppose that there is some transformation T_n, $U = h(T_n)$, such that U has a symmetric distribution. Suppose also that we can obtain the confidence interval of level $1-\alpha$ to $\phi = h(\theta)$. According to Davison and Hinkley (1997), we can use bootstrapping to obtain an approximation of the distribution of $T_n - \theta$ using the distribution of $T_n^* - t$. Thus, we estimate the p quantile of $T_n - \theta$ by the $(J+1)p$-th ordered value of $t^* - t$ estimated by $t^*_{((J+1)p)} - t$ where J is the number of bootstrap samples. Similarly, the p quantile of $h(T_n) - h(\theta) = U - \phi$ can be estimated by the $(J+1)p$-th ordered value of $h(T_n^*) - h(t) = u^* - u$. Let b_p be the p-th quantile of $U - \phi$. Since U has a symmetrical distribution, then $U - \phi$ also has a symmetrical distribution, as soon as it is true that $b_{\frac{\alpha}{2}} = -b_{1-\frac{\alpha}{2}}$. Based on the symmetry of $U - \phi$, we have $h(\ell_{\alpha/2}) = u + b_{\alpha/2}$ and $h(\ell_{1-\alpha/2}) = u + b_{1-\alpha/2}$. As $b_{\alpha/2}$ and $b_{1-\alpha/2}$ are quantiles of $U - \phi$ and we can obtain these quantiles, the lower and upper limits of the confidence intervals are given by $u + (u^*_{((J+1)\alpha/2)} - u)$ and $u + (u^*_{((J+1)(1-\alpha/2))} - u)$, respectively, which lead to the limits

$$u^*_{((J+1)\alpha/2)},\quad u^*_{((J+1)(1-\alpha/2))},$$

whose transformation to θ are given by

$$t^*_{(J+1)\alpha/2},\quad t^*_{(J+1)(1-\alpha/2)}. \tag{8}$$

Note that we do not know the transformation h. The confidence interval of level $1-\alpha$ for the parameter θ does not involve h and it can be evaluated without knowledge of this transformation. The interval (8) is known as the bootstrap interval percentile. According to (Davison and Hinkley 1997, p. 203), the percentile method can be applied to any statistic.

INTERVAL DOUBLE BOOTSTRAP PERCENTILE

When using the percentile method, we can obtain a coverage different than the desired level $(1-\alpha)$. The interesting thing is that we can continue making use of bootstrap to correct this discrepancy. This fact reveals the flexibility of the bootstrap method.

The idea to obtain more accurate confidence intervals is to make use of diagrams double bootstrap, i.e., for each replica of the original bootstrap there will be conducted another bootstrap. Consider the situation where only a trust boundary is of interest and it is the upper limit with nominal confidence level $1-\alpha$, where

$$\Pr\{T_n - \theta \leq a_\alpha(F) \mid F\} = \Pr\left\{t(\widehat{F}_n) - t(F) \leq a_\alpha(F) \;\middle|\; F\right\} = \alpha.$$

We ignore the errors of simulation, which is really evaluated is the confidence limit $t(\widehat{F}_n) - a_\alpha(\widehat{F}_n)$. The bias of the bootstrap percentile follows from the fact that $a_\alpha(\widehat{F}_n) \neq a_\alpha(F)$, which implies

$$\Pr\left\{t(F) \leq t(\widehat{F}_n) - a_\alpha(\widehat{F}_n) \;\middle|\; F\right\} \neq 1 - \alpha.$$

Davison and Hinkley (1997) proposed to correct the bias by adding a correction to $a_\alpha(\widehat{F}_n)$, However, an approach more successful is to adjust the index α. Thus, we can replace $a_\alpha(\widehat{F}_n)$ by $a_{q(\alpha)}(\widehat{F}_n)$ and estimate which the fitted value $(\widehat{q}_\alpha)$ must be used. Therefore, we obtain $q(\alpha)$ satisfying

$$\Pr\left\{t(F) \leq t(\widehat{F}_n) - a_{q(\alpha)}(\widehat{F}_n) \;\middle|\; F\right\} = 1-\alpha. \tag{9}$$

Note that the solution $q(\alpha)$ depends on F, i.e., $q(\alpha) = q(\alpha, F)$. Since the distribution F is unknown, we can estimate $q(\alpha)$ by $\widehat{q}(\alpha) = q(\alpha, \widehat{F}_n)$. Let $x_n^* = \{X_1^*, X_2^*, \ldots, X_n^*\}$ by a sample obtained randomly with replacement from x_n and $x_n^{**} = \{X_1^{**}, X_2^{**}, \ldots, X_n^{**}\}$ a new sample obtained by refitting x_n^* with empirical distribution functions given by $\widehat{F}_n^*$ and $\widehat{F}_n^{**}$, respectively. Let T_n^* and T_n^{**} be the statistics T_n evaluated at x_n^* and x_n^{**}, where t^* and t^{**} are the estimates, respectively. We denote $\Pr^*\{\cdot\}$ as the conditional probability in $\widehat{F}_n$ and $\Pr^{**}\{\cdot\}$ as the conditional probability in $\widehat{F}_n^*$. We can obtain $\widehat{q}(\alpha)$ using the bootstrap version of equation (9) given by

$$\Pr^*\left\{t(\widehat{F}_n) \leq t(\widehat{F}_n^*) - a_{\widehat{q}(\alpha)}(\widehat{F}_n^*) \;\middle|\; \widehat{F}_n\right\} = 1-\alpha. \tag{10}$$

From the definition (10), a scheme involving a second level of bootstrap is given by

$$\Pr^*\left[\Pr^{**}\left\{T_n^{**} \leq 2T_n^* - t \;\middle|\; \widehat{F}_n^*\right\} \geq \widehat{q}(\alpha) \;\middle|\; \widehat{F}_n\right] = 1-\alpha. \tag{11}$$

Davison and Hinkley (1997) proved that the coverage $1-\alpha+O(n^{-a})$ is corrected for $1-\alpha+O(n^{-a-1/2})$. For unilateral confidence limits, we have $a=\frac{1}{2}$ or $a=1$. For the bilateral confidence interval, the coverage $1-\alpha+O(n^{-1})$ is corrected to $1-\alpha+O(n^{-2})$.

In general, especially in non-parametric problems, the determination of (11) can not be done exactly. Thus, approximate methods must be used. A basic algorithm is given as follows. Suppose we have J samples of $\widehat{F}_n$ and denote these estimates by $\widehat{F}_{n,1}^*, \ldots, \widehat{F}_{n,j}^*$, where $\widehat{F}_{n,j}^*$ is the j-th empirical distribution function. Set

$$u_j^* = \Pr(T_n^{**} \leq 2t_j^* - t \mid \widehat{F}_{n,j}^*).$$

The values $u_1^*, \ldots, u_j^*$ can be determined by approximation. We generate K samples of $\widehat{F}_{n,j}^*$ and for each of them we obtain the estimated values $t_{j,1}^{**}, \ldots t_{j,k}^{**}$, for $k = 1, 2, \ldots, K$. So,

$$u_{K,j}^* = K^{-1}\sum_{k=1}^{K} I\{t_{j,k}^{**} \leq 2t_j^* - t\},$$

where $I\{A\}$ is the indicator function for an event A. The Monte Carlo version of (11) is given by

$$J^{-1}\sum_{j=1}^{J} I\{u_{K,j}^* \geq \widehat{q}(\alpha)\} = 1-\alpha,$$

where $\widehat{q}(\alpha)$ is the quantile α of $u_{K,j}^*$. The simplest way to obtain $\widehat{q}(\alpha)$ is sorting the values $u_{K,j}^*$, $u_{K,1}^* \leq u_{K,2}^* \leq \cdots \leq u_{K,J}^* \in (0,1)$, and setting $\widehat{q}(\alpha) = u_{K,(\alpha(J+1))}^*$. The $(J+1)\alpha$-th quantile of $u_{K,j}^*$ is used to obtain the corrected quantile of $t_j^* - t$. The double bootstrap percentile bilateral interval algorithm is presented below:

1. For a given sample (original sample) calculate the quantity t;
2. Generate J samples and obtain t_j^*, for $j = 1, \ldots, J$;

3. Generat K new bootstrap samples for each of the J samples in the previous step, and for each one, calculate $t^{**}_{j,k}$, with $k = 1, 2, \ldots, K$;

4. Determine

$$u^*_j = K^{-1} \sum_{k=1}^{K} I\left\{t^{**}_{j,k} \leq 2t^*_j - t\right\},$$

where I is the indicator function;

5. Order vector u^* with J positions and obtain the lower and upper quantile of u^*, which are given, respectively, by $q_{\text{inf}} = u^*_{(J+1)*\alpha/2}$ and $q_{\text{sup}} = u^*_{(J+1)(1-\alpha/2)}$;

6. Order values $t^*_1, t^*_2, \ldots, t^*_J$ (i.e, $t^*_{(1)} \leq t^*_{(2)} \leq \cdots \leq t^*_{(J)}$) and build the confidence interval for the original sample using quantile estimates evaluated in the previous step. Considering the values of the ordered statistics t^*_j, for $j = 1, 2, \ldots, J$, the limits of the interval of level $1 - \alpha$ are given by

$$t^*_{((J+1)\alpha/2)}, \;\; t^*_{((J+1)(1-\alpha/2))}.$$

SIMULATIONS AND HARDWARE USED

The `pso` function is computationally intensive, especially when the objective function involves some data sets. This situation is common when the goal is to maximize a log-likelihood function. The problem arises with great intensity when we use the `pso` function iteratively as is the case of Monte Carlo simulations. The simulations presented are extremely itensive because they will be studied using bootstrap percentile simple and double (two levels of bootstrap) to obtain interval estimates for the NHG parameters. For the case of double bootstrap, each replica Monte Carlo will have a bootstrap and for each bootstrap will be conducted another bootstrap render the simulations impractical in some hardware.

The simulations are performed using the `R` language (version 3.2.0) in the hardware from the National Center for High Performance Processing of São Paulo, Brazil (CENAPAD-SP). The CENAPAD-SP provides a powerful computing environment, based on machines RISC (IBM) and Intel/Itanimum2 (SGI) with operating system based in Unix. The theoretical processing capacity of these two environments totals around 33 TFLOPS beyond of 250 TB external disk. Its use is measured in Service Units, what are accounted as users run commands and process jobs. In particular, the simulations performed make use of the SGI Altix ICE 8400 LX system installed in CENAPAD-SP that have 64 CPU's and 384 cores Intel Xeon Six Core 5680 of 3.33GHz, 1152 GB of RAM and Infiniband interconnect. The theoretical processing capability of the system is approximately 5 TFLOPS.

The jobs that are submitted for processing on SGI Altix ICE 8400 LX are managed and have resources allocated by the PBSPro software (Altair Portable Batch System Professional 11.0.2), which also has the function of managing the queues of jobs submitted for processing in the cluster. The cluster runs the opreracional system SUSE Linux Enterprise Server 11 SP1 (x86_64) kernel 2.6.32.13 64 bits. Access to

the cluster is through a local machine using SSH (Secure Shell), which is part of the suite of protocols TCP/IP that makes secure remote administration of Unix-type servers.

It is possible to use in an efficient way all of the computing resources of SGI Altix ICE 8400 LX running parallel computing by OpenMPI (Message Passing Interface), which is nothing more than a standard for parallel computing in data communication. The use of standard OpenMPI in `R` is through the `Rmpi`, `SNOW`, `doSNOW` and `foreach` libraries available on the license terms GPL-2 (GNU General Public License).

This pattern allows using `R` by dividing each bootstrap in several cores of different nodes in the cluster. Thus, the code executes sequentially until it finds a parallel builder, i.e., there is a flow of execution main (master thread) and when required new threads are triggered to divide the work and, finally, it is performed a join for recovery the results. Divide each replica of bootstrap does not cause problems because each replica bootstrap is mathematically independent of another.

For the simulations we consider 10,000 replicas of Monte Carlo, and for each replica it is carried out two levels of bootstrap percentile. The first level of bootstrap is composed of 500 samples ($J = 500$) and the second level is carried out 250 new samples ($K = 250$). The simulations are extremely intensive since for each Monte Carlo replica, we perform an optimization by PSO algorithm and within each replica of bootstrap (first level of bootstrap) another optimization is performed by PSO algorithm, closing a total of 5 million optimizations. Table I gives the times of the execution of `R` scripts, in hours, on hardware mentioned above. Note that for $n = 500$ in the double bootstrap scheme, some simulations exceeded 11 days.

The performance of interval estimates is evaluated by percentile bootstrap and double percentile bootstrap at a nominal level of 95% for different sample sizes ($n = 20$, $n = 60$, $n = 100$ and $n = 500$). The real model has fixed parameters $\alpha = 1.5$, $\lambda = 1.3$ and $p = 0.5$.

For the first level of bootstrap, the standard errors of the MLEs are obtained by the PSO method. Small errors in the estimates are obtained for different sample sizes as shown in Figure 4. The performance of interval estimates by single and double bootstrap is assessed by the percentage of coverage, i.e., it is considered the percentage of confidence intervals containing the true parameter. It is noted that build interval estimates by single percentile bootstrap is not a good strategy for obtaining random intervals for the model parameters. In all cases, there are much lower coverages for the nominal levels. However, using the second level of bootstrap to correct the estimate obtained by simple percentile bootstrap method represents a good improvement in the interval estimates. Table II gives the improvement in coverage when using the double percentile bootstrap method. It is also possible to notice a small increase in the range of interval estimates by using the second level of bootstrap. Anyway, the small amplitudes do not represent a problem.

TABLE I
Times of executions of Monte Carlo simulations to evaluate the interval estimates obtained by single and double percentile bootstrap.

n	Simple Bootstrap (In hours)	Double Bootstrap (In hours)
20	105.3212	150.4235
60	175.3455	191.9864
100	213.7565	265.5455
500	400.3253	450.4535

TABLE II
Coverage and range of interval estimates by simple bootstrap, double and nominal level of 95%.

		Simple Bootstrap		Double Bootstrap	
n	**Parameters**	**Coverage (%)**	**Range**	**Coverage (%)**	**Range**
	α	65.0327	2.1562	92.3441	3.2432
$n = 20$	λ	62.1665	2.1605	93.4551	2.3423
	p	67.3002	0.9663	94.2138	1.3563
	α	71.1454	0.7446	93.5233	3.2124
$n = 60$	λ	77.7465	2.1454	92.3453	2.3567
	p	66.7213	1.2423	94.4354	2.8433
	α	72.2198	1.4354	92.4345	3.3315
$n = 100$	λ	69.2300	2.1322	94.2130	1.0035
	p	69.4357	0.5493	93.4365	2.1252
	α	83.1233	2.1215	94.7834	3.2123
$n = 500$	λ	71.9011	1.2321	94.3254	3.2122
	p	72.2965	1.1190	93.9342	1.2134

1 - The column for the range refers to the average range.

2 - We consider 10 thousand Monte Carlo replicas.

EMPIRICAL EXAMPLE

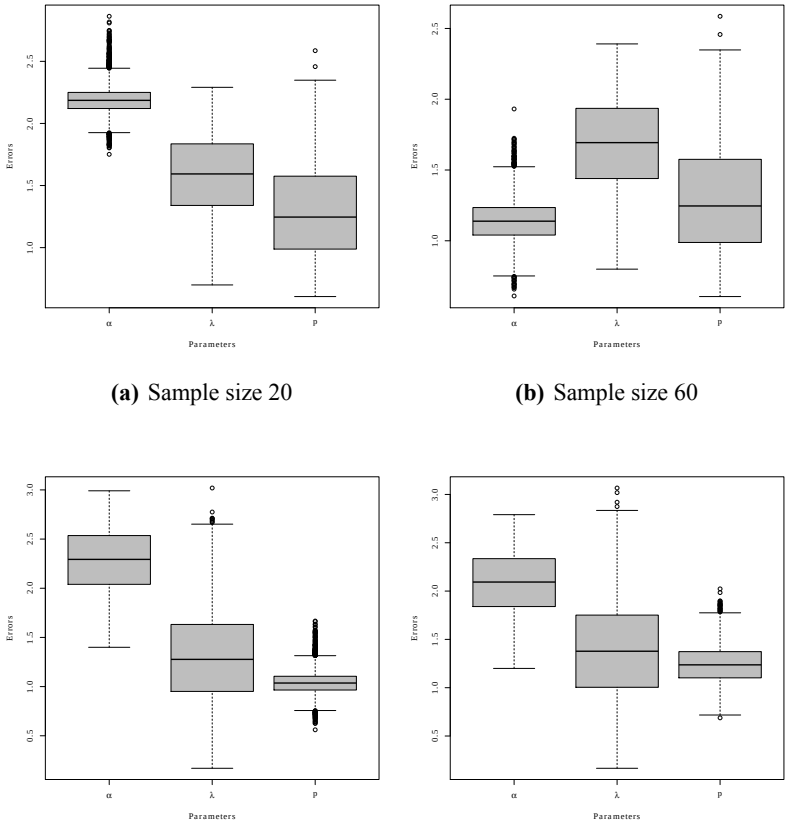
For this application, we use a real data set referred to the times between failures (thousands of hours) of secondary reactor pumps. The data are presented in Table III. A descriptive analysis of the data is given in Table IV.

There are certainly other competitive distributions to fit these data. It is also reasonable to understand that generated distributions can be used in practice and depending on the problem a distribution will provide a better fit than others.

For this application, we consider the NHG(α, λ, p), NH(λ, β) and ENH(α, λ, β) models. The cdfs are given in Table III.

TABLE III
Times between failures (thousands of hours) of secondary reactor pumps.

2.160	0.150	4.082	0.746	0.358	0.199	0.402	0.101	0.605	0.954
1.359	0.273	0.491	3.465	0.070	6.560	1.060	0.062	4.992	0.614
5.320	0.347	1.921							



(a) Sample size 20 **(b)** Sample size 60

(c) Sample size 100 **(d)** Sample size 500

Figure 4 - Errors evaluated by bootstrap and the maximum likelihood using the PSO method with 500 bootstrap replicates.

In order to determine the shape of the most appropriate hazard function for modeling, a graphical analysis data may be performed. In this context, the total time in test (TTT) plot proposed by Aarset (1987) can be used. Let T be a random variable with non-negative values, which represents the survival time. The TTT curve is obtained by constructing the plot of $G(r/n) = [(\sum_{i=1}^{r} T_{i:n}) + (n-r)T_{r:n}]/(\sum_{i=1}^{n} T_{i:n})$ versus r/n, for $r = 1, \ldots, n$, where $T_{i:n}$, $i = 1, \ldots, n$, are the order statistics of the sample [see (Mudholkar and Hutson 1996)].

We use the `R` programming language version (`3.0.2`). In particular, we use the `Adequacy Model` package in version `1.0.8` available for download at http://cran.r-project.org/web/packages/AdequacyModel/index.html under the terms of the GPL-2 and GPL-3. The script was created and is maintained by two authors of this paper. The plots can be easily obtained using the `TTT` function of the `AdequacyModel`. For more details on this function, see `help(TTT)`. The TTT plot for the current data is displayed in Figure 5, which alternates between convex and concave, suggesting, according to Aarset (1987), that the better risk function to the current data has decreasing form.

Figure 6 displays the estimated distributions to the data obtained in a nonparametric manner using kernel density estimation with the Gaussian filter. Let $x_1, \ldots, x_n$ be independent and identically distributed

TABLE IV
Descriptive statistics.

Statistics	Real data sets
	Time between failures (hours)
Mean	1.5779
Median	0.6140
Mode	0.5000
Variance	3.7275
Skewness	1.3643
Kurtosis	0.5445
Maximum	6.5600
Minimum	0.0620
n	23

observations, where each of them follows an unknown f distribution. The kernel density estimator is given by

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

where $K(\cdot)$ is the kernel function usually symmetrical, $\int_{-\infty}^{\infty} K(x)dx = 1$, and $h > 0$ is a smoothing parameter known in the literature as bandwidth. Numerous kernel functions are available in the literature. The standard normal distribution is the most widely used because it has convenient mathematical properties. Silverman (1986) demonstrated that for the K standard normal, the ideal bandwidth is $h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\,\hat{\sigma}\,n^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the sample.

The `AdequacyModel` package provides a computational support for researchers who want to work with continuos distributions, mainly in the survival analysis area. The `AdequacyModel` package is used to evaluate some adequacy statistics such as the AIC (Akaike Information Criterion), CAIC (Consistent Akaikes Information Criterion), BIC (Bayesian Information Criterion), HQIC (Hannan-Quinn information criterion), KS (Kolmogorov-Smirnov), A^* (Anderson-Darling), W^* (Camér-von Misses) and KS (Kolmogorov-Smirnov) statistics. The last two measures are described by Chen and Balakrishnan (1995). The `goodness.fit` is the function in the `AdequacyModel` package used to evaluate these statistics. Other details can be obtained with the command `help(goodness.fit)`.

Chen and Balakrishnan (1995) corrected statistics W^* and A^* based on the suggestions from Stephens (1986). We use these statistics, where we have a random sample $(x_1, \ldots, x_n)$ with empirical distribution

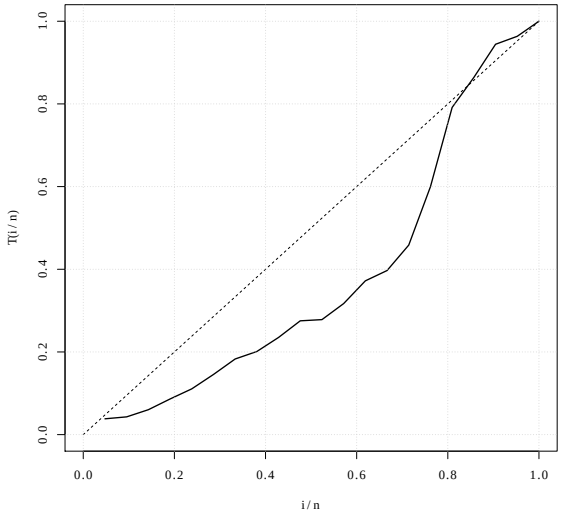


Figure 5 - The TTT plot for the time between failures (thousands of hours) of secondary reactor pumps.

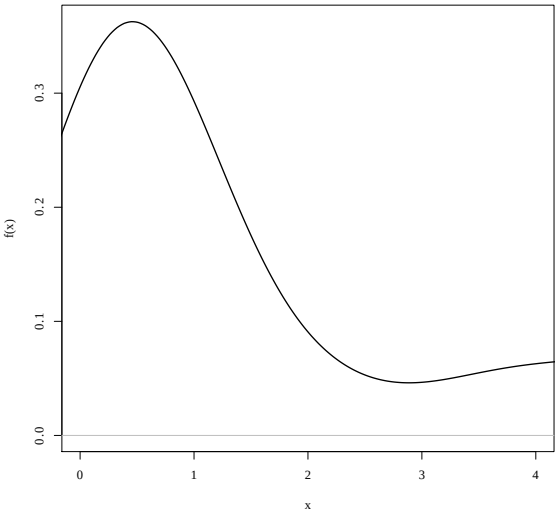


Figure 6 - Gaussian kernel density estimation for the data of the time between failures (thousands of hours) of secondary reactor pumps.

function $F_n(x)$ and want to test if the sample comes from a specified distribution. The statistics W^* and A^* are given by

$$\begin{aligned} W^* &= \left\{ n \int_{-\infty}^{+\infty} \{F_n(x) - F(x;\widehat{\theta}_n)\}^2 dF(x;\widehat{\theta}_n) \right\} \left(1 + \frac{0.5}{n}\right) = W^2 \left(1 + \frac{0.5}{n}\right), \\ A^* &= \left\{ n \int_{-\infty}^{+\infty} \frac{\{F_n(x) - F(x;\widehat{\theta}_n)\}^2}{\{F(x;\widehat{\theta})(1 - F(x;\widehat{\theta}_n))\}} dF(x;\widehat{\theta}_n) \right\} \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right) \\ &= A^2 \left(1 + \frac{0.75}{n} + \frac{2.25}{n^2}\right), \end{aligned}$$

respectively, where $F_n(x)$ is the empirical distribution function, $F(x;\hat{\theta}_n)$ is the specified distribution function evaluated at the MLE $\hat{\theta}_n$ of θ. Note that the statistics W^* and A^* are given by difference distances of $F_n(x)$ and $F(x;\hat{\theta}_n)$. Thus, the lower the statistics values the more evidence we have that $F(x;\hat{\theta}_n)$ generates the sample.

The `R` language is also used to obtain the MLEs by the PSO heuristic method of global optimization discussed in the previous section. Currently, the version `1.0.8` of the `AdequacyModel` package does not support the PSO methodology. However, possibly the standard algorithm or a modification will be present in future versions of the package. The standard errors of the MLEs of the model parameters are evaluated by non-parametric bootstrap. For the calculations, we consider 500 bootstrap samples ($B = 500$) [see (Davison and Hinkley 1997) and (Efron and Tibshirani 1993)].

Let $x_1, \ldots, x_n$ be a random sample. Let also $F_\theta(x)$ be the distribution function of the sample, where θ (unknown) is the true parameter and $\hat{\theta}$ a estimator of θ. A bootstrap sample (non-parametric bootstrap) is obtained sampling with replacement from the original sample. The procedure generates a new sample

$(x_1^*, \ldots, x_n^*)$ (bootstrap sample). Let $\hat{\theta}^*$ be the estimate of θ based on the bootstrap sample and B be the fixed number of bootstrap samples. The bootstrap estimate of the standard error of $\hat{\theta}$ is given by

$$\widehat{\text{se}}(\hat{\theta}) = \sqrt{\frac{1}{B}\sum_{i=1}^{B}(\hat{\theta}_i^* - \bar{\hat{\theta}}^*)} \ ,$$

where,

$$\bar{\hat{\theta}}^* = \frac{1}{B}\sum_{i=1}^{B}\hat{\theta}^*.$$

Table V presents the MLEs of the parameters of the distributions obtained by the PSO method. They are also given for the estimated standard errors obtained via nonparametric bootstrap. Also, we construct interval estimates by double percentile bootstrap for the model parameters. The estimates are presented in Table VI for a confidence level of 95%. We consider 500 samples for the first level and 250 bootstrap samples for the second level. Roughly speaking, all the competing models are able to fit the reactor pumps data. However, the adequacy statistics showed in Table VII indicate that the NHG model yields a better fit to these data than the other models.

TABLE V
The MLEs of the parameters for the fitted NHG, ENH, NH and EXP models through of the PSO method (bootstrap standard errors in parentheses).

Distributions	Estimates of the parameters			Equation
NHG(α, λ, p)	0.4195	5.7294	-0.7929	$F(x) = \dfrac{1-\mathrm{e}^{1-(1+\lambda x)^{\alpha}}}{1-p\,\mathrm{e}^{1-(1+\lambda x)^{\alpha}}}$
	(0.1153)	(2.7918)	(1.2611)	
ENH(α, λ, β)	0.2856	42.0728	3.1569	$F(x) = \left[1-\mathrm{e}^{1-(1+\lambda x)^{\alpha}}\right]^{\beta}$
	(0.4808)	(19.0491)	(1.2583)	
NH(α, λ)	0.4934	2.5010		$F(x) = 1-\mathrm{e}^{1-(1+\lambda x)^{\alpha}}$
	(0.3493)	(1.4762)		

TABLE VI
Interval estimates by double percentile bootstrap for the model parameters at a confidence level of 95%.

Distributions	Interval estimates		
NHG(α, λ, p)	(0.3578; 0.8177)	(0.8884; 13.8570)	$(-3.7684; -0.0001)$
ENH(α, λ, β)	(0.2621; 0.7341)	(0.8400; 49.9999)	(0.9062; 5.8346)
NH(α, λ)	(0.3804; 1.4111)	(0.3665; 5.9659)	

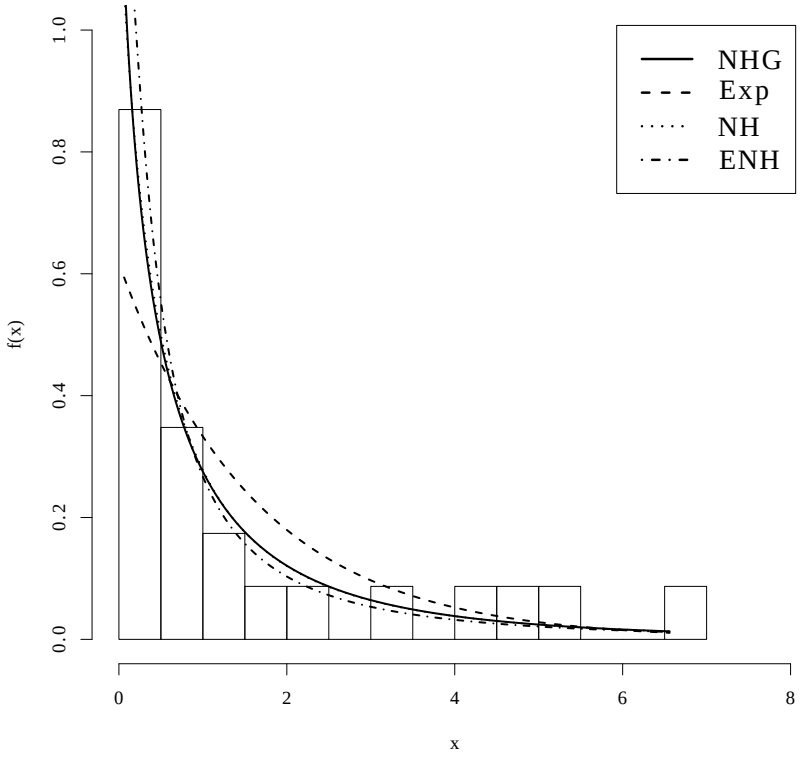


Figure 7 - Estimated for models fitted to the times between failures (thousands of hours) of secondary reactor pumps.

TABLE VII
Goodness-of-fit statistics.

	Statistics						
Distribution	**AIC**	**CAIC**	**BIC**	**HQIC**	**KS**	**A***	**W***
NHG(α, λ, p)	72.8293	74.0925	76.2358	73.6860	**0.1684**	**0.2547**	**0.0323**
ENH(α, λ, β)	289.7261	290.9892	293.1326	290.5828	0.6630	0.3020	0.0424
NH(α, λ)	239.9434	240.5434	242.2144	240.5146	0.7119	0.3546	0.0513

1 - The statistics are obtained in the `AdequacyModel` package version (`1.0.8`).

CONCLUDING REMARKS

We study the NHG distribution, which can be viewed as an improved extension of the Nadarajah-Haghighi (NH) distribution. The NHG density function can take various forms depending on its shape parameters. Moreover, its failure rate function can be decreasing, increasing, upside-down bathtub, bathtub-shaped, constant and decreasing-increasing-decreasing. Then, it can be used quite effectively in analyzing real data in practice. The estimation of the parameters is approached by the maximum likelihood method. The new

distribution may attract wider applications for modeling failure times due to fatigue and lifetime data in fields such as engineering, finance, economics, and insurance, among several others. Two applications to real data illustrate the potentiality of the family. Future research could be addressed to study the complementary Nadarajah-Haghighi-geometric distribution. This class of distributions is a suitable model in a complementary risk problem based in the presence of latent risks, which arise in several areas such as public health, actuarial science, biomedical studies, demography and industrial reliability.

ACKNOWLEDGEMENTS

We would like to thank two referees for insightful comments and suggestions leading to improvements in the article. The financial support from the Brazilian governmental institutions Fundação de Amparo à Ciência e Tecnologia de Pernambuco (FACEPE) , Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) is gratefully acknowledged.

AUTHOR CONTRIBUTIONS

Bourguignon, M. and Silva, R.B. conceived the proposed idea and developed the theory. Marinho, P.R.D. performed the computations, which include the applications to real data. Cordeiro, G.M. supervised the findings of this work. All authors discussed the results and contributed to the final manuscript.

REFERENCES

AARSET MV. 1987. How to identify a bathtub hazard rate. IEEE T Reliab 36: 106-108.

ADAMIDIS K, DIMITRAKOPOULOU T AND LOUKAS S. 2005. On an extension of the exponential-geometric distribution. Stat Probabil Lett 73: 259-269.

ADAMIDIS K AND LOUKAS S. 1998. A lifetime distribution with decreasing failure rate. Stat Probabil Lett 39: 35-42.

BARRETO-SOUZA W, SANTOS AH AND CORDEIRO GM. 2010. The beta generalized exponential distribution. J Stat Comput Simul 80: 159-172.

BASU AP AND KLEIN JP. 1982. Some recent results in competing risks theory. Lecture Notes-Monograph Series 2: 216-229.

BOURGUIGNON M, SILVA RB AND CORDEIRO GM. 2014. A new class of fatigue life distributions. J Stat Comput Simul 84: 2619-2635.

CHAHKANDI M AND GANJALI M. 2009. On some lifetime distributions with decreasing failure rate. Comput Stat Data Anal 53: 4433-4440.

CHEN G AND BALAKRISHNAN N. 1995. A general purpose approximate goodness-of-fit test. J Qual Technol 27: 154-161.

CORDEIRO GM AND DE CASTRO M. 2011. A new family of generalized distributions. J Stat Comput Simul 81: 883-898.

CORDEIRO GM, ORTEGA EM AND NADARAJAH S. 2010. The Kumaraswamy Weibull distribution with application to failure data. J Franklin I 347: 1399-1429.

DAVISON A AND HINKLEY D. 1997. Bootstrap methods and their application. Volume 1. Cambridge: Cambridge University Press.

EFRON B. 1979. Bootstrap methods: another look at the jackknife. The Annals of Statistics 7: 1-26.

EFRON B AND TIBSHIRANI R. 1993. An introduction to the bootstrap. Volume 57. New York: CRC press.

EUGENE N, LEE C AND FAMOYE F. 2002. Beta-normal distribution and its applications. Commun Stat Theory Methods 31: 497-512.

GOMES AE, DA-SILVA CQ, CORDEIRO GM AND ORTEGA EM. 2014. A new lifetime model: the Kumaraswamy generalized Rayleigh distribution. Journal of Statistical Computation and Simulation 84: 290-309.

GUPTA RC, GUPTA PL AND GUPTA RD. 1998. Modeling failure time data by Lehman alternatives. Commun Stat Theory Methods 27(4): 887-904.

KENNEDY J, KENNEDY JF AND EBERHART RC. 2001. Swarm intelligence. Morgan Kaufmann.

LEE C, FAMOYE F AND OLUMOLADE O. 2007. Beta-Weibull distribution: some properties and applications to censored data. J Mod Appl Stat Methods 6: 17.

MAHMOUDI E AND JAFARI AA. 2012. Generalized exponential-power series distributions. Comput Stat Data Anal 56: 4047-4066.

MARSHALL AW AND OLKIN I. 1997a. A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. Biometrika 84: 641-652.

MARSHALL AW AND OLKIN I. 1997b. A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families. Biometrika 84: 641-652.

MORAIS AL AND BARRETO-SOUZA W. 2011. A compound class of Weibull and power series distributions. Comput Stat Data Anal 55: 1410-1425.

MUDHOLKAR GS AND HUTSON AD. 1996. The exponentiated Weibull family: some properties and a flood data application. Commun Stat Theory Methods 25: 3059-3083.

NADARAJAH S AND HAGHIGHI F. 2011. An extension of the exponential distribution. Statistics 45: 543-558.

NADARAJAH S AND KOTZ S. 2006. The exponentiated type distributions. Acta Applicandae Mathematica 92: 97-111.

PRUDNIKOV P, BRYCHKOV A AND MARICHEV OI. 1986. Integrals and series: special functions. Volume 2. CRC Press.

SILVA GO, ORTEGA EM AND CORDEIRO GM. 2010. The beta modified Weibull distribution. Lifetime Data Anal 16: 409-430.

SILVA RB, BOURGUIGNON M, DIAS CR AND CORDEIRO GM. 2013. The compound class of extended Weibull power series distributions. Comput Stat Data Anal 58: 352-367.

SILVA RB AND CORDEIRO GM. 2015. The Burr XII power series distributions: A new compounding family. Brazilian Journal of Probability and Statistics 29(3): 565-589.

SILVERMAN BW. 1986. Density estimation for statistics and data analysis. Volume 26. CRC press.

STEPHENS MA. 1986. Tests based on EDF statistics. Goodness-of-Fit Techniques, RB D'Agostino and MS Stephens, Eds Marcel Dekker .

A - CODE IN R LANGUAGE FOR THE PSO METHOD

```
pso <- function(func,S=150,lim_inf,lim_sup,e=0.0001,data=NULL,N=100){
b_lo = min(lim_inf)
b_up = max(lim_sup)
integer_max = .Machine$integer.max

if(length(lim_sup)!=length(lim_inf)){
   stop("The vectors lim_inf and lim_sup must have the same dimension.")
}
dimension = length(lim_sup)
swarm_xi = swarm_pi = swarm_vi = matrix(NA,nrow=S,ncol=dimension)

# The best position of the particles.
g = runif(n=dimension,min=lim_inf,max=lim_sup)

# Objective function calculated in g.
f_g = func(par=as.vector(g),x=as.vector(data))

if(NaN%in%f_g==TRUE || Inf%in%abs(f_g)==TRUE){
   while(NaN%in%f_g==TRUE || Inf%in%abs(f_g)==TRUE){
      g = runif(n=dimension,min=lim_inf,max=lim_sup)
      f_g = func(par=g,x=as.vector(data))
   }
}

# Here begins initialization of the algorithm.
x_i = mapply(runif,n=S,min=lim_inf,max=lim_sup)

# Initializing the best position of particularities i to initial position.

swarm_pi = swarm_xi = x_i
f_pi = apply(X=x_i,MARGIN=1,FUN=func,x=as.vector(data))

is.integer0 <- function(x){
   is.integer(x) && length(x)==0L
}

if(NaN%in%f_pi==TRUE || Inf%in%abs(f_pi)){
   while(NaN%in%f_pi==TRUE || Inf%in%abs(f_pi)){
      id_inf_fpi = which(abs(f_pi)==Inf)
      if(is.integer0(id_inf_fpi)!=TRUE){
     f_pi[id_inf_fpi] = integer_max
      }
      id_nan_fpi = which(f_pi==NaN)
      if(is.integer0(id_nan_fpi)!=TRUE){
     x_i[id_nan_fpi,] = mapply(runif,n=length(id_nan_fpi),min=lim_inf,
     max=lim_sup)
     swarm_pi = swarm_xi = x_i
     f_pi = apply(X=x_i,MARGIN=1,FUN=func,x=as.vector(data))
      }
   }
}
```

```
minimo_fpi = min(f_pi)
if(minimo_fpi < f_g) g = x_i[which.min(f_pi),]

# Initializing the speeds of the particles.

swarm_vi = mapply(runif,n=S,min=-abs(rep(abs(b_up-b_lo),dimension)),
max=abs(rep(abs(b_up-b_lo),dimension)))

# Here ends the initialization of the algorithm

omega = 0.5
phi_p = 0.5
phi_g = 0.5

m=1
vector_f_g <- vector()

while(is.na(var(vector_f_g)) || m<50 ||
   var(vector_f_g[length(vector_f_g):(length(vector_f_g)-10)])>e){
   # r_p and r_g are randomized numbers in (0.1).
   r_p = runif(n=dimension,min=0,max=1)
   r_g = runif(n=dimension,min=0,max=1)

   # Updating the vector speed.
   swarm_vi = omega*swarm_vi+phi_p*r_p*(swarm_pi-swarm_xi)+
   phi_g*r_g*(g-swarm_xi)

   # Updating the position of each particle.
   swarm_xi = swarm_xi+swarm_vi

   myoptim = function(...) tryCatch(optim(...), error = function(e) NA)

   f_xi = apply(X=swarm_xi,MARGIN=1,FUN=func,x=as.vector(data))
   f_pi = apply(X=swarm_pi,MARGIN=1,FUN=func,x=as.vector(data))
   f_g =  func(par=g,x=as.vector(data))

   if(NaN%in%f_xi==TRUE || NaN%in%f_pi==TRUE){
      while(NaN%in%f_xi==TRUE){
      id_comb = c(which(is.na(f_xi)==TRUE),which(is.na(f_pi)==TRUE))
      if(is.integer0(id_comb)!=TRUE){
         new_xi = mapply(runif,n=length(id_comb),min=lim_inf,
         max=lim_sup)
         swarm_pi[id_comb,]=swarm_xi[id_comb,] = new_xi
         if(length(id_comb)>1){
            if_xi[id_comb] = apply(X=swarm_xi[id_comb,],MARGIN=1,
            FUN=func,x=as.vector(data))
            f_pi[id_comb] = apply(X=swarm_pi[id_comb,],MARGIN=1,FUN=func,
            x=as.vector(data))
         }else{
         f_xi[id_comb] = func(par=new_xi,x=as.vector(data))
         }
      }
```

```
      }
   }

   if(Inf%in%abs(f_xi)==TRUE){
      f_xi[which(is.infinite(f_xi))]=integer_max
   }
   if(Inf%in%abs(f_pi)==TRUE){
      f_pi[which(is.infinite(f_pi))]=integer_max
   }

   # There are values below the lower limit of restrictions?
   id_test_inf=
   which(apply(swarm_xi<t(matrix(rep(lim_inf,S),dimension,S)),1,sum)>=1)
   id_test_sup=
   which(apply(swarm_xi>t(matrix(rep(lim_sup,S),dimension,S)),1,sum)>=1)

   if(is.integer0(id_test_inf)!=TRUE){
      swarm_pi[id_test_inf,] = swarm_xi[id_test_inf,] =
      mapply(runif,n=length(id_test_inf),
      min=lim_inf,max=lim_sup)
   }

   if(is.integer0(id_test_sup)!=TRUE){
      swarm_pi[id_test_sup,] = swarm_xi[id_test_sup,] =
      mapply(runif,n=length(id_test_sup),
      min=lim_inf,max=lim_sup)
   }

   if(is.integer0(which((f_xi<=f_pi)==TRUE))){
      swarm_pi[which((f_xi<=f_pi)),] = swarm_pi[which((f_xi<=f_pi)),]
   }

   if(f_xi[which.min(f_xi)] <= f_pi[which.min(f_pi)]){
      swarm_pi[which.min(f_pi),] = swarm_xi[which.min(f_xi),]
      if(f_pi[which.min(f_pi)] < f_g) g = swarm_pi[which.min(f_pi),]
   } # Here ends the block if.

   vector_f_g[m] = f_g
   m = m+1
   if(m>N){
      break
   }

   } # Here ends the block while.

   f_x = apply(X=swarm_xi,MARGIN=1,FUN=func,x=as.vector(data))
   list(par_pso=g,f_pso=vector_f_g)

} # Here ends the function.
```