

Accurate Mixed Weibull Distribution Fitting by Differential Evolution

Pavel Krömer
Department of Computer Science,
VSB–Technical University of Ostrava
17. listopadu 2172/15
Ostrava, Czech Republic 708 00
pavel.kromer@vsb.cz

Jana Heckenbergerová
Inst. of Mathematics and Quantitative
Methods, University of Pardubice
Studentská 95
Pardubice, Czech Republic 532 10
jana.heckenbergerova@upce.cz

Petr Musilek
Dept. of Electrical and Computer
Engineering, University of Alberta
9211-116 Street NW
Edmonton, Alberta, Canada T6G 1H9
pmusilek@ualberta.ca

ABSTRACT

Mixed Weibull distribution is a probability distribution noted for its wide applicability in many diverse fields. The ability to accurately estimate mixed distribution parameters is essential for data-driven modeling, simulation, and analysis of the phenomena represented by mixed Weibull models. Nature-inspired metaheuristics for continuous parameter optimization have shown good potential for approximating parameters of complex statistical models. Differential evolution is a popular evolutionary real-parameter optimization method with good results in many areas. This work uses differential evolution to fit mixed Weibull distribution to data and analyzes the ability of different differential evolution variants to estimate mixture parameters.

CCS CONCEPTS

•**Theory of computation** → Evolutionary algorithms; •**Computing methodologies** → Bio-inspired approaches; •**Computing methodologies** → Continuous models;

KEYWORDS

Mixed Weibull distribution, Differential Evolution, parameter estimation, distribution fitting, experiments

ACM Reference format:

Pavel Krömer, Jana Heckenbergerová, and Petr Musilek. 2017. Accurate Mixed Weibull Distribution Fitting by Differential Evolution. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017*, 8 pages. DOI: <http://dx.doi.org/10.1145/3071178.3071290>

1 INTRODUCTION

Weibull distribution (WD) is a continuous probability distribution internationally popularized by W. Weibull [9, 11, 16]. It is regarded as one of the most popular models in modern statistics [16] known for its use in reliability engineering and analysis [9, 11] and very large number of applications in many diverse areas [9, 16]. WD and its variants have been successfully used in a wide number of

fields including material science, engineering, physics, chemistry, meteorology and hydrology, biology, medicine, maintenance and replacement [16], economics, complex system reliability analysis [11] and others.

Weibull distribution is an extreme value probability distribution with a number of interesting properties. The traditional WD can be formulated using a 3 or 2-parameter probability density function (*pdf*). The parameters of the *pdf* represent its *location* (α), *scale* (β), and *shape* (γ). The 2-parameter *scale–shape* WD variant, employed in reliability analysis, assumes that the location parameter, α , is equal to zero and uses the scale and shape parameters only [16]. WD has a number of variants and extensions including truncated, log, exponential [16], inverse, reflected [9], and mixed Weibull distributions [9, 16]. Mixed WDs are frequently utilized in applied statistical science to model heterogeneous populations composed of two or more distinct subpopulations [16].

There is a number of different methods for WD parameter estimation. They are based on graphical analysis (empirical cumulative distribution plot, Weibull probability plot, hazard rate plot [9]) the method of moments, minimum–distance, maximum likelihood, least–square estimation, Bayesian approaches [16], and e.g. interval methods [11]. An accurate estimation of mixed WD parameters is, however, still considered a complex problem [16] and optimization approaches are often used [12]. This work introduces a new nature-inspired metaheuristic optimization–based method for mixed WD parameter estimation. It uses the differential evolution (DE) algorithm to find parameters of finite WD mixtures and evaluates the ability of three popular DE variants to discover accurate parameter estimates. The results show that DE is able to find accurate parameters of mixed WDs and is therefore suitable for real–world applications that require exact analysis and/or modelling of phenomena statistically expressed as mixed WDs.

The rest of this paper is organized in the following way. Section 2 introduces Weibull distribution, finite mixtures of probability distributions, and outlines traditional methods for finite WD mixture fitting to empirical data. Differential evolution and its variants, considered in this study, are presented in section 3. The details of the proposed DE for mixed WD fitting are provided in section 4 and experimental evaluation of the proposed approach on a synthetic data set is described in section 5. Finally, the work is concluded in section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '17, Berlin, Germany

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-4920-8/17/07...\$15.00
DOI: <http://dx.doi.org/10.1145/3071178.3071290>

2 MIXED WEIBULL DISTRIBUTION

Simple scale–shape Weibull distribution is defined by a probability density function

$$f(x | \beta, \gamma) = \frac{\beta}{\gamma} \left(\frac{x}{\gamma}\right)^{\beta-1} \cdot \exp\left(-\left(\frac{x}{\gamma}\right)^\beta\right), \quad (1)$$

where $\beta > 0$ and $\gamma > 0$ are distribution scale and shape parameters, respectively [9, 11, 16]. The *pdf* of a finite mixture of n probability distributions is given by

$$f(x | \mathbf{w}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{i=1}^n w_i f_i(x | \beta_i, \gamma_i), \quad (2)$$

where $\mathbf{w} = (w_1, \dots, w_n)$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n)$, and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n)$ are vectors of mixture component parameters. It is required that $w_i > 0$, $\beta_i > 0$, and $\gamma_i > 0$ for all $i \in \{1, \dots, n\}$ and $\sum_{i=1}^n (w_i) = 1$. The *pdf* of a mixed Weibull distribution is then defined by

$$f(x | \mathbf{w}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{i=1}^n w_i \left[\frac{\beta_i}{\gamma_i} \left(\frac{x}{\gamma_i}\right)^{\beta_i-1} \cdot \exp\left(-\left(\frac{x}{\gamma_i}\right)^{\beta_i}\right) \right]. \quad (3)$$

2.1 Parameter estimation

Fitting of mixed WDs to data is a complex problem that involves accurate estimation of mixture parameters [12, 16]. Traditional methods for mixed distribution parameter estimation include graphical and analytical (numerical) approaches such as the method of moments, maximum likelihood estimation, and Bayesian approaches [16]. They very often require solid a priori knowledge of the fitted data or the phenomena it describes and knowledge of the number of subpopulations. The procedure for parameter estimation is then chosen with respect to expected data properties. Various hybrid [5], optimization–based [12] and nature–inspired [1, 4, 7, 8, 19] approaches have been proposed to tackle the drawbacks of the traditional mixed WD fitting methods.

Advanced methods for mixed WD fitting have different forms. For example, a hybrid method for WD mixture parameter estimation [5] applies Bayes’ theorem to split product failure data into several groups according to posterior probability that a failure belongs to certain subpopulation. Then, mixture parameters are estimated and amended by least–square optimization. Another algorithm [12] combines rough parameter estimation from data histogram with maximum likelihood estimation of component shape parameters. The parameters of mixed WD components are in this approach optimized independently. Nonlinear least–squares regression, implemented by a quasi–Newton procedure, is another algorithm that was employed to fit mixtures of 2 WDs in [10].

Nature–inspired metaheuristics are used to approximate parameters of complex statistical models in a growing number of cases. They are increasingly popular due to their accuracy, ability to fit statistical models to data with different properties, and because of the lack of prior knowledge about the modelled phenomena they usually require [1, 4, 7, 8, 19].

2.2 Nature–inspired estimation of probability distribution parameters

Many different nature–inspired metaheuristics have been employed to optimize probability distribution and mixed probability distribution parameters. Simulated annealing [1] was used to optimize parameters of a 3–parameter WD and genetic algorithms were applied to estimate parameters of a mixed WD [7]. Krohling et al. [8] used a simple variant of particle swarm optimization (PSO), called bare bones particle swarm optimization, to estimate parameters of a mixture of 2 WDs. A later work [19] adapted the same algorithm to fit Weibull distribution to censored data. Another recent study [4] applied PSO to approximate parameters of a finite mixture of circular normal (von Mises) probability distribution.

The overview of recent applications of nature–inspired metaheuristics to estimation of probability distribution and mixed probability distribution parameters clearly shows that there is need for accurate, robust, and adaptive distribution fitting methods. This work uses differential evolution, a popular and successful real–parameter optimization metaheuristic, to fit mixed Weibull distribution to data. It defines an encoding of mixed distribution parameters and evaluates the ability of three variants of DE to optimize distribution parameters.

3 DIFFERENTIAL EVOLUTION

The DE is a popular stochastic evolutionary optimization algorithm that evolves a population of real encoded vectors representing the solutions to given problem. It was introduced by Storn and Price in 1995 [17, 18] and it quickly became a popular alternative to the more traditional types of evolutionary algorithms. The algorithm evolves a population of candidate solutions by iterative modification of candidate solutions by the application of the differential mutation and crossover [14]. In each iteration, so called trial vectors are created from current population by the differential mutation and further modified by various types of crossover operator. At the end, the trial vectors compete with existing candidate solutions for survival in the population.

The DE starts with an initial population of M real–valued vectors. The vectors are initialized with real values either randomly or so, that they are evenly spread over the problem space and provide more a complete initial coverage of the problem space. The latter initialization leads to better results of the optimization [14]. During the optimization, the DE generates new vectors that are scaled perturbations of existing population vectors. The algorithm perturbs selected base vectors with the scaled difference of two (or more) other population vectors in order to produce the trial vectors. The trial vectors compete with members of the current population with the same index called the target vectors. If a trial vector represents a better solution than the corresponding target vector, it takes its place in the population [14].

The two most significant parameters of the DE are scaling factor and mutation probability [14]. The scaling factor, $F \in [0, \infty]$, controls the rate at which the population evolves and the crossover probability, $C \in [0, 1]$, determines the ratio of elements that are transferred to the trial vector from its opponent. The size of the population and the choice of operators are important parameters of the optimization process.

The basic operations of the classic DE can be summarized using the following formulae [14]: the random initialization of the i th vector with N parameters is defined by

$$x_j^i = \text{rand}(b_j^L, b_j^U), \quad j \in \{1, \dots, N\}, \quad (4)$$

where b_j^L is the lower bound of j -th parameter, b_j^U is the upper bound of j -th parameter, and $\text{rand}(a, b)$ is a function generating a random number from the range $[a, b]$. A simple form of the standard differential mutation, *DE/rand/1*, is given by

$$\mathbf{v}^i = \mathbf{v}^{r1} + F(\mathbf{v}^{r2} - \mathbf{v}^{r3}), \quad (5)$$

where F is the scaling factor and \mathbf{v}^{r1} , \mathbf{v}^{r2} , and \mathbf{v}^{r3} are three random vectors from the population. The vector \mathbf{v}^{r1} is the base vector, \mathbf{v}^{r2} and \mathbf{v}^{r3} are the difference vectors, and \mathbf{v}^i is the trial vector. It is required that $i \neq r1 \neq r2 \neq r3$.

An alternative differential mutation which favours exploitation over exploration, *DE/best*, is defined by

$$\mathbf{v}^i = \mathbf{x}^{\text{best}} + F(\mathbf{v}^{r1} - \mathbf{v}^{r2}) \quad (6)$$

and combines two randomly chosen difference vectors with the best vector in population, \mathbf{x}^{best} .

The uniform (binomial) crossover that combines the target vector, \mathbf{x}^i , with the trial vector, \mathbf{v}^i , is given by

$$v_j^i = \begin{cases} v_j^i & \text{if } \text{rand}(0, 1) < C \text{ or } j = j_{\text{rand}} \\ x_j^i & \text{otherwise} \end{cases} \quad (7)$$

for each $j \in \{1, \dots, N\}$. The random index j_{rand} is in the above selected randomly as $j_{\text{rand}} = \text{rand}(1, N)$. The uniform crossover replaces the parameters in \mathbf{v}^i by the parameters from the target vector \mathbf{x}^i with probability $1 - C$. The outline of the traditional DE according to [3, 14] is summarized in Algorithm 1. The DE is a

Algorithm 1: A summary of traditional Differential Evolution

```

1 Initialize the population  $P$  consisting of  $M$  vectors using Equation (4);
2 Evaluate an objective function ranking the vectors in the population;
3 while Termination criteria not satisfied do
4   Let  $G$  = number of current generation;
5   for  $i \in \{1, \dots, M\}$  do
6     Differential mutation: Create trial vector  $\mathbf{v}^i$  according to Equation (5);
7     Validate the range of coordinates of  $\mathbf{v}^i$ . Optionally adjust coordinates
       of  $\mathbf{v}^i$  so, that it is valid solution to given problem;
8     Perform uniform crossover. Select randomly one parameter  $j_{\text{rand}}$  in  $\mathbf{v}^i$ 
       and modify the trial vector using Equation (7);
9     Evaluate the trial vector.;
10    if trial vector  $\mathbf{v}^i$  represent a better solution than target vector  $\mathbf{x}^i$  then
11      | add  $\mathbf{v}^i$  to  $P^{G+1}$ 
12    else
13      | add  $\mathbf{x}^i$  to  $P^{G+1}$ 
14    end
15  end
16 end

```

successful evolutionary algorithm designed for continuous parameter optimization driven by the idea of scaled vector differentials. That makes it an interesting alternative to the wide spread genetic algorithms that are designed to work primarily with discrete encoding of the candidate solutions. As well as GA, it represents a highly parallel population based stochastic search metaheuristic.

In contrast to the GA, the differential evolution uses the real encoding of candidate solutions and different operations to evolve the population. It results in different search strategy and different directions found by DE when crawling a fitness landscape of the problem domain.

The traditional DE has shown an ability to solve a wide range of problems. However, its performance in particular domains strongly relies on the selection of differential mutation and crossover operators as well as parameters F and C [15]. A number of self-adaptive DE variants was designed to mitigate this dependence. Among them, the Self-Adaptive Differential Evolution (SaDE) algorithm has shown good results for many types of tasks [2, 15]. Another interesting parameter-free DE variant is called Gaussian Bare-bones Differential Evolution (GBDE) [20].

3.1 Self-Adaptive Differential Evolution

SaDE combines the principles of probabilistic trial vector generation strategy selection, scaling factor randomization, and crossover probability adaptation.

In each generation, G , SaDE selects for every target vector, \mathbf{x}^i , a trial vector generation strategy, s_k , from a pool of predefined strategies, $S = \{s_1, s_2, \dots, s_K\}$. The strategy is selected according to strategy selection probability, $p_{k,G}$, that reflects the historical performance of each strategy from S . The strategy selection probability is adapted on the basis of the number of successes (i.e. the number of times the trial vector, \mathbf{v}^i , is better solution than the target vector, \mathbf{x}^i) and failures (i.e. the number of times \mathbf{v}^i is worse solution than \mathbf{x}^i) of trial vectors generated by s_k during a fixed number of past generations known as *learning period* (LP). The algorithm employs success and failure memories, SM and FM , that store the number of successes, $ns_{k,g}$, and failures, $nf_{k,g}$, of each strategy in the past LP generations.

The strategy selection probabilities are in each generation, G , $G > LP$, given by

$$p_{k,G} = \frac{S_{k,G}}{\sum_k S_{k,G}}, \quad (8)$$

$$S_{k,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{k,g}}{\sum_{g=G-LP}^{G-1} ns_{k,g} + \sum_{g=G-LP}^{G-1} nf_{k,g}} + \epsilon, \quad (9)$$

where ϵ is a small constant (here, $\epsilon = 0.01$) employed to tackle cases with zero success rate [15]. Initial strategy selection probabilities are for the first LP generations set to be equal, i.e. $p_{k,G} = \frac{1}{K}$, $k \in \{1, 2, \dots, K\}$.

Trial vector generation strategies can include arbitrary combinations of differential mutation and crossover. The strategies used in this study are summarized in fig. 1. The strategies *DE/rand/1/bin* and *DE/rand/2/bin* have slow convergence but strong exploration capability. *DE/rand-to-best/2/bin* has fast convergence, especially for unimodal problems, but tends to get trapped in local optima and suffers from premature convergence. *DE/current-to-rand/1* is a rotation invariant type of DE that has good efficiency for rotated problems [15].

DE/rand/1/bin:

$$v_j^i = \begin{cases} v_j^{r1} + F(v_j^{r2} - v_j^{r3}), & \text{if } \text{rand}(0, 1) < C \text{ or } j = j_{\text{rand}} \\ x_j^i, & \text{otherwise} \end{cases}, \quad (10)$$

DE/rand-to-best/2/bin:

$$v_j^i = \begin{cases} x_j^i + F(x_j^{\text{best}} - x_j^i) + F(v_j^{r1} - v_j^{r1}) + F(v_j^{r3} - v_j^{r4}), & \text{if } \text{rand}(0, 1) < C \text{ or } j = j_{\text{rand}} \\ x_j^i, & \text{otherwise} \end{cases}, \quad (11)$$

DE/rand/2/bin:

$$v_{i,j}^t = \begin{cases} v_j^{r1} + F(v_j^{r2} - v_j^{r3}) + F(v_j^{r4} - v_j^{r5}), & \text{if } \text{rand}(0, 1) < C \text{ or } j = j_{\text{rand}} \\ x_j^i, & \text{otherwise} \end{cases}, \quad (12)$$

DE/current-to-rand/1:

$$v^i = x_j^i + \kappa(v^{r2} - x^i)F(v^{r2} - v^{r3}), \quad (13)$$

where x^{best} is the best solution found so far and $\kappa = \text{rand}(0, 1)$ is a random parameter generated for each trial vector, v^i .

Figure 1: SaDE trial vector generation strategies.

Scaling factor F is in SaDE for each trial vector selected randomly from normal distribution with mean 0.5 and standard deviation 0.3 [15]

$$F_{i,G} = \mathcal{N}(0.5, 0.3). \quad (14)$$

Scaling factors drawn from such distribution fall in 99.7% of cases into the range $[-0.4, 1.4]$ and enable both, exploitation (small F) and exploration (large F) [15].

The value of crossover probability is an important problem-dependent parameter that has a major impact on algorithm performance [15]. In SaDE, crossover probability is for each application of a particular trial vector generation strategy, s_k , generated randomly according to

$$C_{k,G} = \mathcal{N}(C_{m_{k,G}}, 0.1), \quad (15)$$

where $C_{m_{k,G}}$ is mean of the (normal) random distribution of C s for strategy s_k in generation G . The values of crossover probabilities used in the past LP successful applications of each trial vector generation strategy are stored in so called *Cm memory* (**CmM**). Self-adaptive differential evolution is appealing because it alleviates the need to tune algorithm parameters.

3.2 Gaussian Bare-bones Differential Evolution

GBDE extends the original Bare-bones Differential Evolution algorithm [13] which was inspired by a particle swarm optimization (PSO) variant called Bare-bones PSO [6]. Bare-bones PSO, based on theoretical analysis of PSO convergence, eliminates particle velocity and samples new position of a particle, i , from Gaussian distribution with mean and standard deviation defined by the best position visited by the swarm, y , and best position visited by the particle, y_i^t so that new particles are centered around weighted

average of y and y_i^t . The exploration/exploitation ratio is automatically adjusted from an initial focus on exploration to later focus on exploitation [20].

GBDE applies similar principles within the framework of DE. It uses Gaussian mutation strategy defined by

$$v^i = \mathcal{N}(\mu, \sigma), \quad (16)$$

where \mathcal{N} is a normal (Gaussian) random distribution with mean μ and standard deviation σ ,

$$\mu = \frac{x^{\text{best}} + x^i}{2}, \quad \sigma = |x^{\text{best}} - x^i|, \quad (17)$$

where x^{best} is the best solution found so far and x^i is the target vector.

GBDE employs DE's traditional binomial crossover. To avoid the need to choose crossover probability, C , manually, it uses the following self-adaptive strategy for its dynamic selection

$$C_{G+1}^i = \begin{cases} C_G^i, & \text{if } f_{\text{obj}}(v^i) \leq f_{\text{obj}}(x^i) \\ \mathcal{N}(0.5, 0.1), & \text{otherwise} \end{cases}, \quad (18)$$

where C_G^i is crossover probability associated with i th target vector, x^i , in generation G , and $\mathcal{N}(0.5, 0.1)$ is a random value sampled from normal distribution with mean 0.5 and standard deviation 0.1. The strategy attempts to change the crossover probability C^i every time it did not generate a better solution than the target vector x^i . Due to its stochastic nature, Gaussian mutation prefers exploration over exploitation. Modified GBDE algorithm (MGBDE) [20] chooses between Gaussian mutation according to (16) and *DE/best* mutation, defined by (6), at random. Bare-bones variants of differential evolution are interesting for mixed WD parameter estimation also because Bare-bones PSO was recently used for fitting of 2-component mixtures of WDs with success [8].

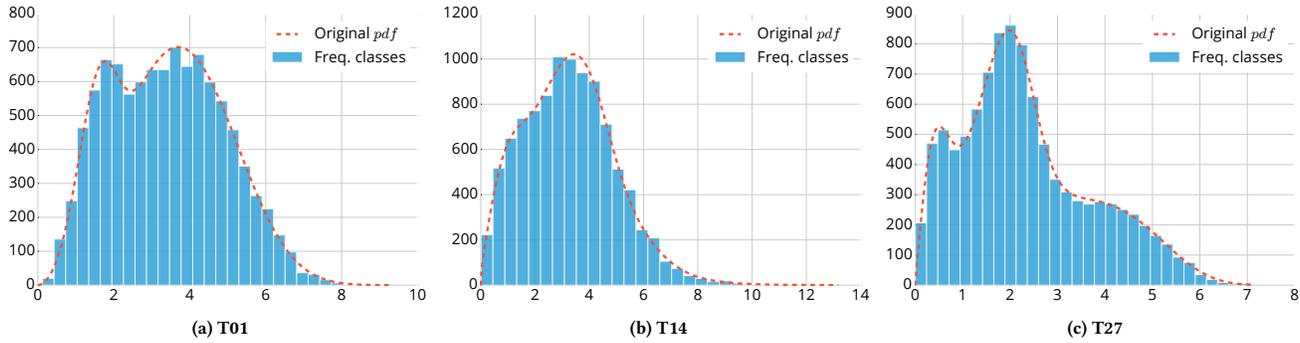


Figure 2: Examples of test data sets.

4 DE FOR MIXED WEIBULL DISTRIBUTION PARAMETER ESTIMATION

Real-valued candidate solution representation (encoding) and suitable fitness function have to be defined in order to use DE and its variants for mixed WD parameter estimation.

This work uses a parameter encoding approach introduced by Heckenbergerova et al. [4] to fit mixed von Mises distribution to meteorological (wind direction) data by PSO. Because the DE has similar real-valued nature as PSO, identical solution representation can be used. The mixture of n Weibull distributions is represented by a candidate vector $\mathbf{v} = (v_1, \dots, v_N), v_i \in [0, 1], N = 3n$, composed of three parts encoding the vectors of mixed WD parameters, $(\mathbf{w}, \boldsymbol{\beta}, \boldsymbol{\gamma})$, respectively

$$\mathbf{v} = (\underbrace{v_1, \dots, v_n}_{\mathbf{w}}, \underbrace{v_{n+1}, \dots, v_{2n}}_{\boldsymbol{\beta}}, \underbrace{v_{2n+1}, \dots, v_N}_{\boldsymbol{\gamma}}). \quad (19)$$

The decoding of $\boldsymbol{\beta}$ involves scaling of $v_i, i \in \{n + 1, \dots, 2n\}$ to $[0, \beta_{\max}]$, and the decoding of $\boldsymbol{\gamma}$ requires scaling of $v_i, i \in \{2n + 1, \dots, N\}$ to $[0, \gamma_{\max}]$. The upper bounds of scale and shape parameters, β_{\max} and γ_{\max} , have to be chosen with respect to precision of the numerical method used to evaluate the probability density function (3).

To satisfy the requirement for finite probability distribution mixtures, $\sum_{i=1}^n (w_i) = 1$, the following decoding rule has been devised for component weights

$$w_j = \begin{cases} v_i, & \text{if } j = 1, \\ v_i \cdot \left(1 - \sum_{l=1}^{j-1} w_l\right), & \text{otherwise.} \end{cases} \quad (20)$$

This guarantees that the sum of component weights is equal to 1 and does not impose any additional constraints on candidate vector handling. All vectors created during the optimization process are therefore valid candidate solutions representing a finite mixture of n Weibull distributions.

The fitness of each candidate solution is evaluated using root mean-squared error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{i=1}^T (O_i - np_i)^2}, \quad (21)$$

where T is the number of frequency classes, O_i is the observed frequency of i -th class, n is the sum of all observed frequencies, and p_i is the theoretical (modelled) probability of i -th frequency class. RMSE is a common measure often used to evaluate the differences between predicted and observed values. It combines intuitive interpretation with good mathematical properties [21]. Low value of RMSE suggests good fit of observed and theoretical probabilities while large values of RMSE are associated with loose correspondence between observed and modeled phenomena.

5 EXPERIMENTS

A series of computational experiments was conducted to study the ability of the traditional DE, SaDE, and MGBDE to estimate mixed WD parameters and fit it to data. In order to establish an evaluation testbed, 30 different random test data sets were generated. Each test data set consisted of 10,000 random samples drawn from a finite mixture of 3, 6, or 9 Weibull distributions with randomized scale and shape parameters. The number of frequency classes, T , was set to 30. A visual illustrations of selected test data sets is shown in fig. 2. Although the data is artificial, the randomized procedure used to obtain it makes it similar to real-world data sets describing e.g. annual wind speed distribution in specific locations.

All three investigated DE variants were implemented in C++ and used to fit a finite mixture of a maximum of 12 WDs to the test data sets. The algorithms were executed with the following fixed parameters, determined on the basis of best practices, previous experience, and extensive trial-and-error runs: the population size, M , was set to 100 and the scaling factor, F , and the crossover probability, C , of the traditional DE were fixed to 0.9. RMSE was used as fitness function. The maximum number of generations was 10,000 and the maximum number of fitness function evaluations was 1,000,000. The stopping criterion is in line with the main purpose of the experiment: an empirical comparison of selected DE variants. The traditional DE, used in the experiments, was the *DE/rand/1* version of the algorithm.

Table 1: Final RMSE of mixed Weibull distributions fitted by different DE variants.

expe- riment	DE			SaDE			MGBDE		
	Min	Mean (σ)	Max	Min	Mean (σ)	Max	Min	Mean (σ)	Max
T01	5.402	6.524 (0.591)	7.741	3.511	7.953 (1.751)	11.677	6.639	7.940 (0.894)	10.620
T02	6.890	7.261 (0.421)	8.465	4.873	9.051 (1.559)	12.895	7.079	8.614 (0.892)	11.422
†T03	6.743	7.652 (0.524)	8.375	2.419	7.675 (2.562)	11.584	6.275	7.363 (0.708)	9.087
†T04	5.138	5.388 (0.138)	5.635	2.148	4.595 (1.954)	8.877	3.795	5.606 (0.612)	6.746
T05	11.695	12.206 (0.136)	12.316	10.505	13.880 (2.241)	19.228	12.319	12.477 (0.103)	12.731
†T06	8.818	11.127 (5.843)	33.317	5.438	13.210 (6.034)	27.595	9.022	9.982 (0.657)	11.029
T07	12.146	12.670 (0.404)	13.121	12.966	14.107 (1.558)	18.408	12.282	13.996 (1.294)	18.224
T08	5.880	6.287 (0.305)	7.414	6.126	8.369 (2.077)	13.299	6.088	6.967 (0.939)	9.728
T09	8.586	8.824 (0.880)	13.571	6.540	13.404 (7.549)	48.369	8.944	9.758 (1.059)	13.102
T10	4.458	6.438 (0.746)	8.871	7.122	10.313 (2.216)	14.812	4.376	7.241 (1.303)	9.433
T11	5.577	7.427 (1.181)	9.222	2.556	13.598 (6.584)	37.725	5.995	8.666 (2.490)	14.664
T12	3.766	3.835 (0.110)	4.234	3.877	10.847 (9.098)	35.862	3.872	6.013 (1.022)	8.448
T13	0.747	0.799 (0.076)	1.065	0.638	4.074 (3.170)	12.025	0.935	1.680 (0.736)	4.327
T14	7.682	9.915 (0.617)	10.280	4.446	12.611 (5.580)	23.582	9.391	10.567 (0.797)	13.305
T15	6.907	7.617 (0.205)	7.818	7.568	11.961 (3.491)	22.004	7.542	8.986 (1.421)	12.588
T16	4.532	5.332 (0.480)	5.858	4.888	8.785 (5.003)	22.419	4.901	6.654 (0.817)	9.069
T17	4.083	4.743 (0.235)	5.154	3.269	6.223 (2.929)	12.178	3.662	5.206 (0.697)	8.123
T18	5.973	6.681 (0.332)	7.587	6.221	9.337 (2.557)	18.420	4.992	9.039 (1.675)	12.172
T19	10.727	11.836 (0.471)	12.402	1.936	14.495 (5.202)	26.115	11.496	12.562 (0.460)	13.786
T20	6.651	7.914 (0.587)	10.276	7.340	8.888 (1.496)	11.933	7.511	8.218 (0.381)	8.855
T21	10.003	10.404 (0.485)	11.887	9.589	14.179 (5.110)	32.788	10.193	12.054 (1.135)	15.585
T22	2.645	3.536 (0.516)	4.478	1.561	5.178 (2.616)	12.223	3.997	4.636 (0.344)	5.542
T23	1.874	3.090 (0.443)	4.211	3.737	9.336 (7.332)	31.592	2.848	5.500 (2.662)	15.242
T24	3.526	3.894 (0.189)	4.179	3.211	6.613 (3.549)	19.177	3.955	5.088 (1.064)	8.107
T25	4.302	4.811 (0.351)	5.899	5.113	9.441 (5.170)	24.163	5.169	7.114 (1.224)	8.808
T26	2.968	4.795 (0.921)	5.988	1.094	7.260 (4.347)	18.822	2.655	6.464 (1.603)	10.727
T27	3.782	4.066 (0.306)	4.571	3.450	8.757 (7.716)	28.153	3.892	5.252 (0.572)	6.560
†T28	4.338	6.724 (5.389)	36.093	3.503	9.233 (3.748)	16.150	5.849	6.791 (0.438)	7.854
T29	1.639	2.393 (0.358)	3.277	1.763	5.757 (5.423)	24.578	1.805	2.914 (1.321)	8.034
T30	6.244	6.730 (0.490)	8.009	5.359	8.638 (4.041)	22.817	6.321	7.789 (1.293)	10.829

The RMSE of WD mixtures, fitted to test data sets by DE, SaDE, and MGBDE, was recorded and analyzed. Because the investigated algorithms are stochastic, all experiments were executed 30 times and the reported results are averages of the 30 independent runs.

A summary of experimental results is provided in table 1. It clearly shows that the traditional DE is the most successful of the investigated algorithms in terms of *average* final root mean error after 10,000 generations. However, SaDE obtained for most test data sets better final solution in the *best* run. MGBDE obtained the best final solution for two out of 30 test data sets and was outperformed by DE and SaDE in all other cases. A statistical analysis of experimental results revealed that the difference between best and second best algorithm was statistically significant (at significance level $\alpha = 0.05$) in 26 out of 30 test cases. The test cases, where the difference between final results was not significant, are in table 1 marked with †. It can be seen that this situation applies only to test cases where the average DE result was not the best (i.e. for test data sets T03, T04, and T06). That means that the DE was either

best or the differences between results obtained by investigated algorithms were not statistically significant at $\alpha = 0.5$.

The evolution of average RMSE of WD mixtures with parameters optimized by DE, SaDE, and MGBDE in time is illustrated in fig. 3 and fig. 4. Figure 3 shows average RMSE of the investigated algorithms for three test cases where the difference between final RMSE was statistically significant while fig. 4 illustrates the evolution of average RMSE for three test cases where the difference in final RMSE was statistically insignificant. The plots also show for each investigated algorithm a 95% confidence interval for the average RMSE in every generation. They illustrate that the DE has in most cases obtained solutions with the best average final fitness but at the same time show that it is outperformed by SaDE and MGBDE during the first approx. 500 generations. Similar behaviour was observed in all test cases.

Finally, the best WD mixtures, evolved for sample test data sets by the DE, are shown in fig. 5. The plots demonstrate that the evolved mixtures correspond to the frequency classes in the test

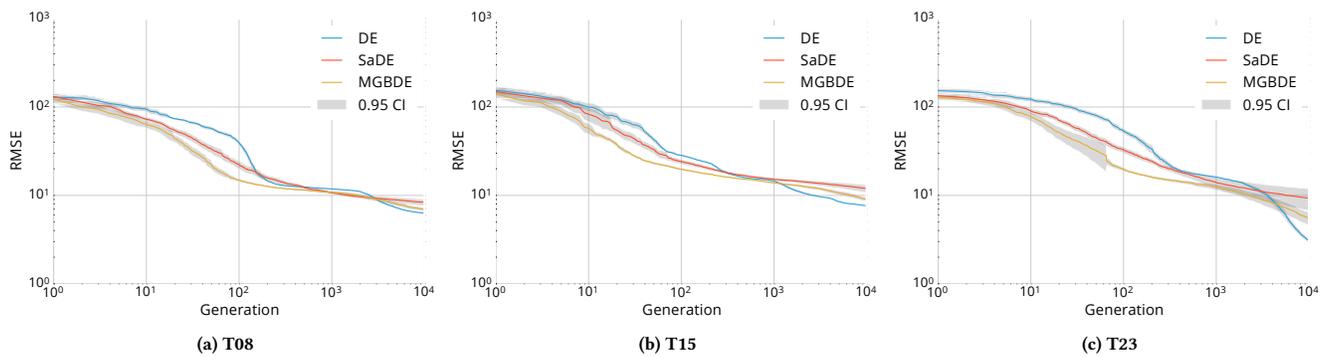


Figure 3: Examples of average RMSE of mixed WDs with parameters optimized by DE, SaDE, and MGBDE in time for test cases with statistically significant difference in final fitness at significance level $\alpha = 0.05$. The displayed range corresponds to a 95% confidence interval for average fitness values. Both axes in all plots have logarithmic scale.

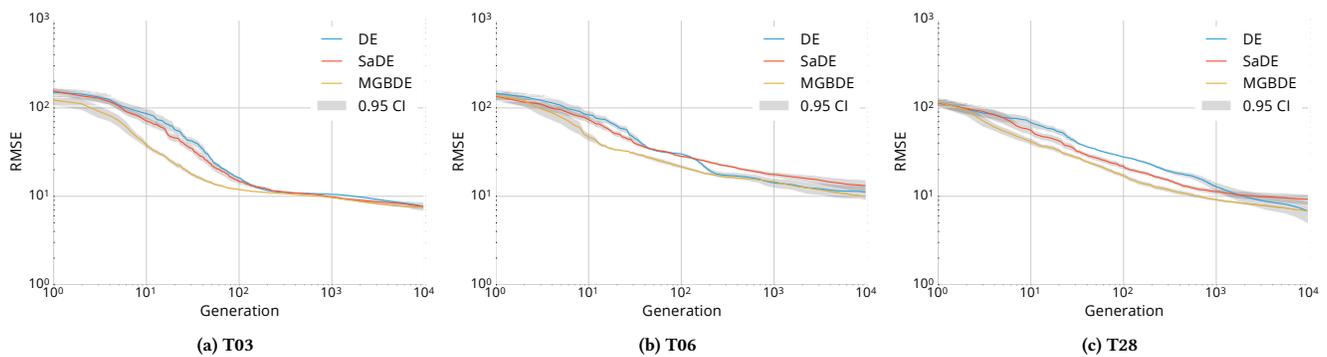


Figure 4: Examples of average RMSE of mixed WDs with parameters optimized by DE, SaDE, and MGBDE in time for test cases with statistically insignificant difference in final fitness at significance level $\alpha = 0.05$. The displayed range corresponds to a 95% confidence interval for average fitness values. Both axes in all plots have logarithmic scale.

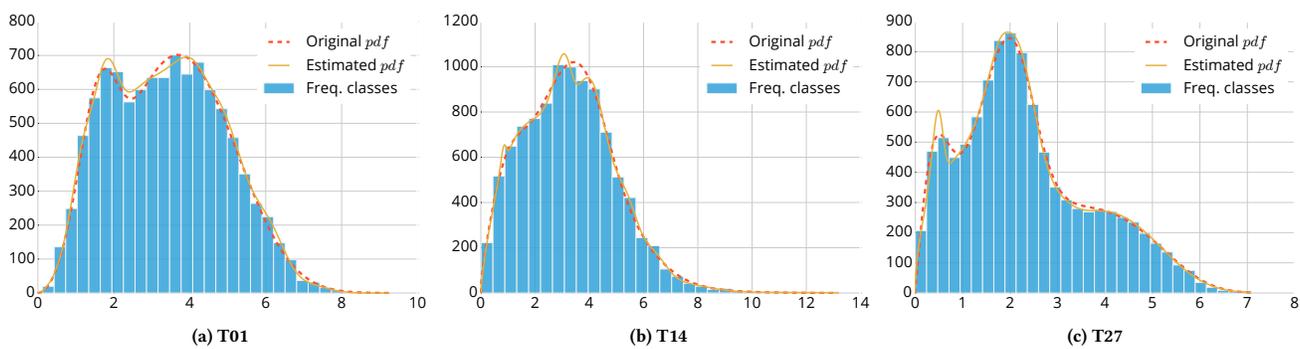


Figure 5: Examples of the best mixed WDs evolved by DE.

data sets very accurately. They also show that they are able to capture the trends (i.e. subpopulations) in the modelled data.

6 CONCLUSIONS

A novel nature-inspired method for accurate fitting of mixed Weibull distributions to empirical data was designed and evaluated in this work. The method does not require any a priori knowledge about

the data and can find parameters for finite mixtures of an arbitrary number of Weibull distributions. The proposed approach represents mixture parameters as a vector of floating point values and defines a decoding procedure that ensures that all candidate vectors, generated throughout the evolution, correspond to valid finite WD mixtures.

The ability of three popular variants of differential evolution to find accurate WD mixture parameters was evaluated experimentally. A series of computational experiments, performed with 30 synthetic test data sets, showed that differential evolution is a suitable metaheuristic strategy for accurate WD mixture fitting to empirical data. Further experiments demonstrated that the traditional *DE/rand/1* algorithm is able to find after 10,000 generations more accurate mixture parameters than self-adaptive differential evolution and modified Gaussian bare-bones differential evolution. However, SaDE and MGBDE are able to deliver better mixture parameters in the first approx. 500 generations of the algorithms. A statistical analysis of experimental results showed that the difference between RMSE of fitted mixtures, obtained by *DE/rand/1* and other considered DE variants, was statistically significant in 26 out of 30 test cases.

Future work in this area will take several directions. First, the proposed encoding will be used for other real-parameter optimization methods (e.g. particle swarm optimization, mean-variance mapping optimization) and their ability to find mixed WD parameters will be assessed. Second, the ability of nature-inspired methods to model data produced by real-world sources such as environmental monitoring (wind speed measurement) and the usefulness of obtained models in context of high-level applications such as distributed stochastic energy generation and integration will be evaluated.

ACKNOWLEDGMENTS

This work was supported by the Czech Science Foundation under the grant no. GJ16-25694Y and in part by the grants of SGS No. SP2017/100 and SP2017/85, VŠB - Technical University of Ostrava, Czech Republic. The support of the Natural Sciences and Engineering Research Council (NSERC) of Canada is gratefully acknowledged as well.

REFERENCES

- [1] Babak Abbasi, Abdol Hamid Eshragh Jahromi, Jamal Arkat, and Mehdi Hosseinkouchack. 2006. Estimating the Parameters of Weibull Distribution Using Simulated Annealing Algorithm. *Appl. Math. Comput.* 183, 1 (Dec. 2006), 85–93. DOI: <http://dx.doi.org/10.1016/j.amc.2006.05.063>
- [2] Saber M. Elsayed, Ruhul A. Sarker, and Daryl Leslie Essam. 2013. An Improved Self-Adaptive Differential Evolution Algorithm for Optimization Problems. *IEEE Trans. Industrial Informatics* 9, 1 (2013), 89–99. DOI: <http://dx.doi.org/10.1109/TII.2012.2198658>
- [3] Andries Engelbrecht. 2007. *Computational Intelligence: An Introduction, 2nd Edition*. Wiley, New York, NY, USA.
- [4] Jana Heckenbergerová, Petr Musilek, and Pavel Krömer. 2015. Optimization of Wind Direction Distribution Parameters Using Particle Swarm Optimization. In *Afro-European Conference for Industrial Advancement*, A. Abraham, P. Krömer, and V. Snasel (Eds.). Advances in Intelligent Systems and Computing, Vol. 334. Springer International Publishing, 15–26. DOI: http://dx.doi.org/10.1007/978-3-319-13572-4_2
- [5] D. B. Kececioglu and Wendai Wang. 1998. Parameter estimation for mixed-Weibull distribution. In *Reliability and Maintainability Symposium, 1998. Proceedings, Annual*. 247–252. DOI: <http://dx.doi.org/10.1109/RAMS.1998.653782>
- [6] J. Kennedy. 2003. Bare bones particle swarms. In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*. 80–87. DOI: <http://dx.doi.org/10.1109/SIS.2003.1202251>
- [7] B. Klein and B. Bertsche. 2010. Estimating parameters of mixed Weibull distributions using genetic algorithms. In *Reliability, Risk and Safety: Back to the Future: Proceedings and Monographs in Engineering, Water and Earth Sciences, ESREL 2010, September 15, 2010, Rhodes, Greece*, B.J.M. Ale, I.A. Papazoglou, and E. Zio (Eds.). Taylor & Francis, 861–868.
- [8] Renato A. Krohling, Mauro Campos, and Patrick Borges. 2010. *Soft Computing in Industrial Applications: Algorithms, Integration, and Success Stories*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter Bare Bones Particle Swarm Applied to Parameter Estimation of Mixed Weibull Distribution, 53–60. DOI: http://dx.doi.org/10.1007/978-3-642-11282-9_6
- [9] C.D. Lai. 2013. *Generalized Weibull Distributions*. Springer Berlin Heidelberg.
- [10] D. Ling, H. Z. Huang, and Yu Liu. 2009. A method for parameter estimation of Mixed Weibull distribution. In *2009 Annual Reliability and Maintainability Symposium*. 129–133. DOI: <http://dx.doi.org/10.1109/RAMS.2009.4914663>
- [11] J.I. McCool. 2012. *Using the Weibull Distribution: Reliability, Modeling and Inference*. Wiley.
- [12] Marko Nagode and Matija Fajdiga. 2000. An improved algorithm for parameter estimation suitable for mixed Weibull distributions. *International Journal of Fatigue* 22, 1 (2000), 75–80. DOI: [http://dx.doi.org/10.1016/S0142-1123\(99\)00112-7](http://dx.doi.org/10.1016/S0142-1123(99)00112-7)
- [13] Mahamed G.H. Omran, Andries P. Engelbrecht, and Ayed Salman. 2009. Bare bones differential evolution. *European Journal of Operational Research* 196, 1 (2009), 128–139. DOI: <http://dx.doi.org/10.1016/j.ejor.2008.02.035>
- [14] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. 2005. *Differential Evolution A Practical Approach to Global Optimization*. Springer-Verlag, Berlin, Germany.
- [15] A.K. Qin, V.L. Huang, and P.N. Suganthan. 2009. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *Evolutionary Computation, IEEE Transactions on* 13, 2 (April 2009), 398–417.
- [16] H. Rinne. 2008. *The Weibull Distribution: A Handbook*. CRC Press.
- [17] R. Storn. 1996. Differential evolution design of an IIR-filter. In *Proceedings of IEEE International Conference on Evolutionary Computation*. 268–273. DOI: <http://dx.doi.org/10.1109/ICEC.1996.542373>
- [18] R. Storn and K. Price. 1995. *Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*. Technical Report.
- [19] Fu-Kwun Wang. 2014. Using BBPSO Algorithm to Estimate the Weibull Parameters with Censored Data. *Communications in Statistics - Simulation and Computation* 43, 10 (2014), 2614–2627. DOI: <http://dx.doi.org/10.1080/03610918.2012.762386>
- [20] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran. 2013. Gaussian Bare-Bones Differential Evolution. *IEEE Transactions on Cybernetics* 43, 2 (April 2013), 634–647. DOI: <http://dx.doi.org/10.1109/TSMCB.2012.2213808>
- [21] I.H. Witten, E. Frank, and M.A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Elsevier Science.