

# Nonconvex Functions Optimization Using an Estimation of Distribution Algorithm Based on a Multivariate Extension of the Clayton Copula

Harold D. de Mello Jr., André V. Abs da Cruz, and Marley M.B.R. Vellasco

Department of Electrical Engineering  
Pontifical Catholic University of Rio de Janeiro  
Rio de Janeiro, RJ, Brazil  
{harold, andrev, marley}@ele.puc-rio.br

**Abstract.** This paper presents a copula-based estimation of a distribution algorithm with parameter updating for numeric optimization problems. This model implements an estimation of a distribution algorithm using a multivariate extension of Clayton's bivariate copula (MEC-EDA) to estimate the conditional probability for generating a population of individuals. Moreover, the model uses traditional mutation and elitism operators jointly with a heuristic for a population restarting in the evolutionary process. We show that these approaches improve the overall performance of the optimization compared to other copula-based EDAs.

**Keywords:** Continuous numeric optimization, evolutionary computation, estimation of distribution algorithms, copulas.

## 1 Introduction

Numerical optimization is an important task that arises in several different knowledge domains. Evolutionary algorithms are capable of finding good solutions with a lower computational cost for different optimization problems. Nevertheless, conventional evolutionary algorithms have difficulty addressing optimization problems when the number of variables, constraints and goals increases. This difficulty relates partly to the association between the performance and the number of parameters, which must be determined for each problem, and to the inability to extract and use knowledge acquired throughout the search process [1].

One possible approach to overcoming these problems is to use Estimation of Distribution Algorithms (EDAs) [2]. EDAs constitute a class of evolutionary algorithms that construct a probability distribution throughout evolution by analyzing the most promising solutions. This probability distribution is used to generate new individuals instead of using mutation and crossover operators as traditional evolutionary algorithms do.

Different EDA models, which differ in how they estimate the probabilistic model, have been proposed, including but not limited to models based on copula. The

concept of copula was introduced by Sklar (1959) [3] and has been extensively used in finance [4]. More recently, copula theory has been applied to Evolutionary Computation [5]. Copula theory allows any multivariate distribution to be expressed as a function (the copula) of its marginal distributions, which describe the behavior of each individual variable; the copula contains the required information to describe the dependency among all variables and is invariant to non-linear transformation [6].

In the case of multidimensional problems, EDA probabilistic models are built, essentially, from bivariate copula [7] because these represent most parametric copulas. Using bivariate copula makes the estimation of the multivariate system easier. In [8], the use of empirical bivariate Archimedean copulas in EDAs was investigated to optimize  $n$ -dimensional problems, while using a constant value for the parameters of the copula.

This study discusses an EDA (MEC-EDA) that is similar to the approach proposed in [8] and applies it to some benchmark functions. Our approach differs in some aspects: the copula parameter is estimated dynamically, using dependency measures; we use information contained in the probability distribution and a classic mutation operator to preserve population diversity; and we use a heuristic to reinitialize the population throughout an elitist evolution. Specifically, MEC-EDA was based on the Clayton's copula because it is simple and no numerical integration is required to compute probabilities. This paper is organized as follows: Section 2 discusses some general aspects regarding copula theory. Section 3 presents copula-based EDAs and our proposed model. Section 4 presents and discusses experimental results on benchmark functions. Section 5 articulates some general conclusions.

## 2 Copula theory

### 2.1 Theorems, Properties and Definitions

A function  $C(u_1, \dots, u_n): [0, 1]^n \rightarrow [0, 1]$  is an  $n$ -dimensional copula if it satisfies the some basic conditions related to boundary conditions and increasing property [3]. Thus, copula  $C$  is a distribution function in  $[0, 1]^n$  that has marginal uniform functions  $u_k$ ,  $k = 1, \dots, n$  in  $(0, 1)$ .

According to Sklar's Theorem [3], a joint Cumulative Distribution Function (CDF)  $F$  of random variables  $X_1, X_2, \dots, X_n$ , with continuous marginal distributions  $F_1, F_2, \dots, F_n$ , respectively, can be characterized by a single  $n$ -dimensional dependency function or copula  $C$ , such that for all vectors  $x \in \mathbb{R}^n$ :

$$F(x_1, x_2, \dots, x_n) = C(F_1(x_1), F_2(x_2), \dots, F_n(x_n)) \quad (1)$$

Similarly, for any vector  $\mathbf{u} \in [0, 1]^n$ ,

$$C(u_1, u_2, \dots, u_n) = F(F_1^{-1}(u_1), F_2^{-1}(u_2), \dots, F_n^{-1}(u_n)) \quad (2)$$

where  $F_i^{-1}(u) = \inf\{x_i \in \mathbb{R}: F_i(x_i) \geq u\}$  for  $i = 1, \dots, n$  is the generalized inverse function of the marginal distribution function  $u \in (0, 1)$ .

Copulas as a modelling dependence tool require algorithms that allow generating random variables. They are based on a definition of conditional distribution. If  $U_1, \dots, U_n$  are described by a joint distribution function  $C$ , then the conditional distribution  $U_k$ , given the values of  $U_1, \dots, U_{k-1}$ , can be determined by:

$$C_k(u_k|u_1, \dots, u_{k-1}) = \frac{\partial^{k-1} C_k(u_1, \dots, u_k)}{\partial u_1 \dots \partial u_{k-1}} \bigg/ \frac{\partial^{k-1} C_{k-1}(u_1, \dots, u_{k-1})}{\partial u_1 \dots \partial u_{k-1}} \quad (3)$$

## 2.2 Archimedean Copulas

Archimedean copulas can represent highly specialized dependence structures. Nevertheless, this type of copula is not directly derived from Sklar's Theorem. For the bivariate case, this copula holds the following representation:

$$C(u_1, u_2) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2)) \quad u_1, u_2 \in [0, 1] \quad (4)$$

where  $\varphi(\cdot)$  is known as a generator function of the copula, and  $\varphi^{-1}$  is its inverse.

Several Archimedean copula families are described in [3]; most of them depend on a single parameter ( $\theta$ ) that controls the dependency structure. However, this work used only one of them in constructing the EDA (Table 1). In this copula, the perfect dependency among the random variables occurs when  $\theta \rightarrow \infty$ , while  $\theta = 1$  indicates total independence among the variables.

**Table 1.** Clayton's bivariate copula

$C(u_1, u_2)$	$\varphi(t)$	$\theta$
$(u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}$	$\theta^{-1}(t^{-\theta} - 1)$	$[0, \infty]$

An additional advantage of Archimedean copulas is that they can be easily used to generate multivariate distributions from extensions of Archimedean 2-copulas. The simplest method used in this work that is named an exchangeable multivariate Archimedean copula (EAC) nevertheless poses a limitation: the dependency structure between any pair of variables is described by the same parameter  $\theta$ , regardless of dimension. In this respect, Archimedean constructions (NACs) and pair-copulas models [9] are more flexible.

## 2.3 Copula Parameter Estimation

The parameter estimation of copulas is usually accomplished using one of the following different approaches: i) maximum likelihood; ii) inference functions for the margins method; or iii) semi-parametric maximum likelihood. An alternative and less computationally intense method that is used in this work is the estimation of moments based on Kendall's tau ( $\tau$ ) [10]. In this method, the relationship between the rank

correlation and the copula's  $\theta$  parameter is used. For Clayton's copula, the following relationships between  $\tau$  and  $\theta$  holds:

$$\tau_{\theta} = \frac{\theta}{\theta+2} \quad (5)$$

### 3 Copula-Based Estimation of Distribution Algorithms (CEDAs)

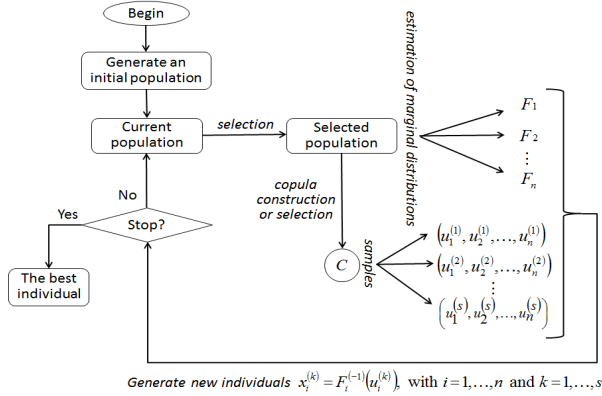
EDAs are a class of optimization algorithms whose most important step and bottle-neck is estimating the joint probability distribution associated with the variables from the most promising solutions determined by the evaluation function. This approach is also the aspect that differs most among EDAs. The complexity of this probabilistic model can be classified into the following categories: independent, pairwise dependent, multivariate dependency and mixed models.

#### 3.1 General Copula-Based EDAs

Recently, a new approach to developing EDAs to solve a real-valued optimization problem has been developed that is based on copula theory. In [5], it is possible to find a review of this research and to find an R package for working with copula-based EDAs.

In copula-based EDAs, the step for estimating the probabilistic model is divided into two parts: i) estimating the marginal distributions,  $F_i$ , for  $i = 1, \dots, n$ , and ii) estimating the dependency structure. Typically, a specific distribution is assumed to be the correct distribution for each marginal distribution, and its parameters are estimated based on maximum likelihood. In other cases, kernel density estimators (KDE), marginal empirical distributions or estimators based on Kendall's inversion were used.

After the marginal distributions are estimated, the selected population is transformed into uniform variables in the interval  $[0, 1]$  by evaluating each cumulative marginal distribution. This transformed population is then used to estimate a copula-based model  $C$  that describes the dependency among the variables. The sampling step creates a population by sampling the dependent distributions  $(u_1^{(k)}, u_2^{(k)}, \dots, u_n^{(k)})$  for  $k = 1, \dots, s$  samples that were created by the copula. New individuals are calculated by  $x_i^{(k)} = F_i^{-1}(u_i)$ , where  $F_i^{-1}$  is the inverse function of  $F_i$ . According to (2) the new individual  $(x_1, x_2, \dots, x_n)$  is a sample that obeys the joint distribution combined by the copula and the marginal distributions. This entire process is repeated until a stop condition is attained, and the best individual is found, as seen in figure 1.



**Fig. 1.** Flowchart of a general copula-based EDA. Adapted from [11].

### 3.2 Proposed Model

The main reason to propose a copula-based EDA is the copulas' flexibility to build effective joint distributions of the search space at the expense of the common and convenient assumption asserted in most EDAs, i.e., that the variables follow a Gaussian distribution. This conjecture often generates inconsistent results and, therefore, a poor performance of the algorithm, particularly in problems of numerical optimization in multivariate spaces. Thus, the use of copulas in EDAs can simplify their operations, after the construction of the probability models is performed in less time and with greater accuracy compared to other estimation techniques.

The first step of MEC-EDA initializes a random population within the search space for the problem. This population is then evaluated, and the correlation matrix for all variables is calculated using the current population as the data inputs. Subsequently, it is possible to calculate Kendall's tau and to obtain the parameter  $\theta$  for Clayton's copula using equation (6). Then, the algorithm simulate pseudo-random uniform variables from a Clayton's copula using the conditional approach described in [4]. This technique is useful for recursively generating observations of random variables uniformly distributed over  $[0, 1]$  whose joint distribution function is a chosen copula. Thus, using the definition (3), the Clayton's bivariate copula and its generator function and the parameter  $\theta$ , the algorithm can generate samples of a  $n$ -variate distribution.

The next step is estimating the marginal distributions. Our model employed empirical marginal distributions. The use of empirical marginals CDF rather than raw data made the experiments comparable, and the rank statistics tend to cope better with real-world systematic biases and errors.

Finally, a new population is obtained through the linear interpolation of the sample  $u$  in the selected population:

$$x = F_i^{-1}(u) = \text{interpolate}(x_i^{<j>}, x_i^{<j+1>}) \quad (6)$$

where  $interpolate(x_i^{<j>}, x_i^{<j+1>})$  denote an interpolated number between two consecutive samples of the sorted current population.

This new population is merged to a mutated version of the older. A new evaluation is performed, and the worst individuals are eliminated. Subsequently, a percentage of individuals obtained from a variation of the learned CDF and with a lower probability of emerging in the evolutionary process (rebel individuals) [12], is added to the final population. The algorithm proceeds with the joint distribution estimation and sampling until a stop condition is satisfied. A pseudocode of this algorithm is given in the following figure.

Whenever the perfect dependency among the random variables occurs ( $\theta \rightarrow \infty$ ), a population's percentage is restarted uniformly in the real interval of search space and the remaining population is restarted uniformly within the limits of the best individual variables at that generation. Another heuristic used is a periodic reset of the limits of the best individual variables according to the limits of search space. All these approaches help to preserve diversity.

```

1. begin
2.    $t \rightarrow 0$ ;
3.   Generate an initial population  $P(t)$  randomly;
4.   while (termination criteria are not satisfied) do
5.     Evaluate  $\Phi(P(t))$ ;
6.     Compute the matrix of Kendall's tau:  $(n, m)$ ;
7.     if (any of  $n$  variables generate the same value in all individuals) do
8.       Restart population  $P(t)$ , retaining only the best individual;
9.     end
10.    Compute Clayton's copula parameter  $\theta$  using Kendall's tau average;
11.    Select the best  $k$  individuals from  $P(t)$ :  $S(P(t))$ ;
12.    Generate samples  $u$  from a multivariate extension of Clayton's bivariate;
13.    Determine the marginal empirical cumulative distributions  $F_i$ ;
14.    Generate  $k$  new individuals:  $x = F_i^{-1}(u) = interpolate(x_i^{<j>}, x_i^{<j+1>})$ ;
15.    Apply a mutation operator in certain individuals from the older population;
16.    Merge the populations (lines 14, 15), evaluate and eliminate the worst individuals;
17.    Add rebel individuals to the final population;
18.     $t \rightarrow t + 1$ ;
19.  end
20. end

```

Fig. 2. MEC-EDA pseudocode

## 4 Experimental Results and Discussion

### 4.1 Benchmark Functions

The performance of the proposed model was evaluated using the traditional benchmark functions: Sphere, Ackley, Rastrigin, Griewank, Rosenbrock and the Summation Cancellation<sup>1</sup>. Sphere, Griewank, Ackley and Rastrigin have their global optimum at  $x=(0, \dots, 0)$ , with evaluation zero; Summation Cancellation has its global optimum at  $x=(0, \dots, 0)$ , with evaluation  $-10^5$ , and Rosenbrock has its global optimum at  $x=(1, \dots, 1)$ , with evaluation zero. The search space was:  $[-600, 600]$  in Sphere and

<sup>1</sup> Modified to a minimization problem.

Griewank,  $[-30, 30]$  in Ackley,  $[-0.16, 0.16]$  in Summation Cancellation,  $[-5.12, 5.12]$  in Rastrigin, and  $[-9, 11]$  in Rosenbrock. All these benchmark functions are nonconvex, except the Sphere and the Summation Cancellation.

All tests were performed with  $n = 10$  variables to be optimized. For each function, 30 independent experiments were performed using two stop criteria: i) reaching the global optimum with a precision greater than  $1e-6$  or ii) reaching 300,000 evaluations. In all experiments, 55% of individuals were restarted uniformly in the real interval of the search space, and 45% of individuals were restarted uniformly in the real interval of the population's maximum and minimum in each respective generation, retaining always the best individual. These limits were reset to the search space in each of the 10 consecutive restarts. The mutation operator was applied in the first 5 individuals (step 15 of pseudocode), with a rate of 70% for Rosenbrock and 100% to all the other functions. Moreover, 5% of rebel individuals were added to the final population. The results are shown in the table 2. For each function, we present results for our model (MEC-EDA) and for the models described in [5]. Each column shows the population size, the best fitness and the worst fitness found, the average final fitness and the number of evaluations that were performed. Different population sizes were evaluated (5, 10, 20, 30, 50, 100 and 200). We selected the minimum population size required by the algorithm to find the global optimum of each function with a high success rate and the minimum number of function evaluations.

**Table 2.** Performance comparison for 10 variables

Sphere Problem							Ackley Problem						
Algorithm	Pop. Size	Success Rate	Best Fitness	Worst Fitness	Average Fitness	N° Eval.	Algorithm	Pop. Size	Success Rate	Best Fitness	Worst Fitness	Average Fitness	N° Eval.
MEC-EDA	5	30/30	2.9e-7	9.9e-7	7.9e-7 ± 1.9e-7	3,102 ± 357	MEC-EDA	5	30/30	7.2e-7	9.9e-7	9.1e-7 ± 8.2e-8	4,900 ± 1,041
UMDA	81	30/30	—	—	6.9e-7 ± 2.3e-7	3,823 ± 128.3	UMDA	81	30/30	—	—	8.1e-7 ± 1.1e-7	4,998 ± 88.0
GCEDA	310	30/30	—	—	6.5e-7 ± 2.0e-7	13,082 ± 221.4	GCEDA	310	30/30	—	—	6.5e-7 ± 2.0e-7	13,082 ± 221.4
CVEDA	104	30/30	—	—	6.7e-7 ± 1.8e-7	4,777 ± 118.8	CVEDA	104	30/30	—	—	8.1e-7 ± 1.0e-7	6,330 ± 163.2
DVEDA	104	30/30	—	—	6.7e-7 ± 2.0e-7	4,787 ± 100.2	DVEDA	104	30/30	—	—	7.9e-7 ± 1.4e-7	6,678 ± 133.8
Cóputa MIMIC	150	30/30	—	—	6.9e-7 ± 1.7e-7	6,495 ± 209.0	Cóputa MIMIC	150	30/30	—	—	8.0e-7 ± 1.2e-7	10,784 ± 143.7

Rastrigin Problem							Griewank Problem						
Algorithm	Pop. Size	Success Rate	Best Fitness	Worst Fitness	Average Fitness	N° Eval.	Algorithm	Pop. Size	Success Rate	Best Fitness	Worst Fitness	Average Fitness	N° Eval.
MEC-EDA	7	30/30	1.9e-7	9.9e-7	7.9e-7 ± 2.0e-7	6,667 ± 1,789	MEC-EDA	5	30/30	1.4e-7	9.9e-7	8.2e-7 ± 1.6e-7	5,919 ± 5,265
UMDA	447	30/30	—	—	6.7e-7 ± 2.3e-7	33,614 ± 2,452	UMDA	111	30/30	—	—	6.6e-7 ± 1.9e-7	5,224 ± 231.2
GCEDA	721	30/30	—	—	6.8e-7 ± 1.8e-7	46,095 ± 2,158	GCEDA	355	30/30	—	—	6.9e-7 ± 1.8e-7	15,099 ± 414.1
CVEDA	447	30/30	—	—	6.6e-7 ± 1.7e-7	32,914 ± 2,011	CVEDA	142	30/30	—	—	7.0e-7 ± 1.8e-7	6,579 ± 389.9
DVEDA	325	30/30	—	—	7.3e-7 ± 1.7e-7	24,710 ± 1,754	DVEDA	150	30/30	—	—	6.5e-7 ± 2.4e-7	6,785 ± 338.1
Cóputa MIMIC	386	30/30	—	—	6.4e-7 ± 2.1e-7	27,315 ± 1,673	Cóputa MIMIC	188	30/30	—	—	6.6e-7 ± 1.8e-7	8,221 ± 220.4

Rosenbrock Problem							Summation Cancellation Problem						
Algorithm	Pop. Size	Success Rate	Best Fitness	Worst Fitness	Average Fitness	N° Eval.	Algorithm	Pop. Size	Success Rate	Best Fitness	Worst Fitness	Average Fitness	N° Eval.
MEC-EDA	30	20/30	5.5e-7	6.5e+0	1.2e+0 ± 2.5e+0	180,805 ± 105,783	MEC-EDA	10	0/30	-99,999.95	-99,996.86	-99,996.12 ± 7.8e-1	300,011 ± 4.0
UMDA	2000	0/30	—	—	8.0e+0 ± 2.6e-2	300,000 ± 0.0	UMDA	2000	0/30	—	—	-5.7e+2 ± 3.4e+2	300,000 ± 0.0
GCEDA	2000	0/30	—	—	7.5e+0 ± 1.9e-1	300,000 ± 0.0	GCEDA	355	30/30	—	—	-1.0e+5 ± 1.3e-7	42,434 ± 305.4
CVEDA	2000	0/30	—	—	7.5e+0 ± 1.1e-1	193,867 ± 48,243	CVEDA	325	30/30	—	—	-1.0e+5 ± 1.3e-7	44,622 ± 858.3
DVEDA	2000	0/30	—	—	7.5e+0 ± 1.5e-1	172,200 ± 35,184	DVEDA	965	30/30	—	—	-1.0e+5 ± 9.3e-8	117,408 ± 959.4
Cóputa MIMIC	2000	0/30	—	—	7.6e+0 ± 1.3e-1	139,000 ± 5,139	Cóputa MIMIC	2000	0/30	—	—	-2.3e+4 ± 2.7e+4	300,000 ± 0.0

Only linear and independence relationships are considered in the models of [5], and all algorithms use normal marginal distributions. Because these models have not restarted the population, they use a third stop criterion: namely, whenever the standard deviation of the evaluation of the solutions in the population is less than  $1e-8$ .

It is possible to verify that the proposed algorithm provides superior results for most functions, thus yielding a smaller number of evaluations to achieve the optimization target. The proposed model failed to optimize Rosenbrock, even though it generated 20 successful experiments and the best average fitness compared to other copula-based EDAs. MEC-EDA also failed to optimize the Summation Cancellation. The algorithms with the best average fitness and number of evaluations are highlighted in the table 2.

## 4.2 Discussion

The results show that, despite the algorithm's simplicity, small improvements can improve the performance of a copula-based EDA. Our approach uses an alternative method for the parameter estimation of the Archimedean copula, as well as strategies related to restarting the population, a traditional mutation operator and rebel individuals to avoid premature convergence and stagnation in local optima.

The preliminary results are good and suggest that with some adjustment, it might be possible to improve the results, particularly for the Rosenbrock function, for which the algorithm was incapable of achieving the global minimum in all experiments and for the Summation Cancellation, for which the average fitness nearly reached the global optimum. A learning strategy to define the proportion of individual restarts and to minimize the number of individuals restarting should solve these failed cases and improve the evaluations number because MEC-EDA works well with a considerably smaller population compared to other CEDAs. Nevertheless, an additional number of benchmark functions must be used to verify the proposed algorithm's performance.

## 5 Conclusion

This paper presented a new model (MEC-EDA) based on Estimation of Distribution Algorithms and copulas to optimize numerical functions. The presented results are promising, although further adjustments and tests must be performed. Specifically, the impact of estimating Kendall's tau on each generation of the EDA must be verified. Additional tests could be performed with different Archimedean copulas to determine whether the type of copula used is important and the results compared to other efficient heuristics, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

**Acknowledgments.** This work has been financially supported by CNPq and FAPERJ.

## References

1. Hauschild, M., Pelikan, M.: An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation* 3(1), 111–128 (2011)
2. Larrañaga, P., Lozano, J.A. (eds.): Estimation of distribution algorithms. A new tool for evolutionary computation. Kluwer Academic Publishers, London (2002)



3. Nelsen, R.B.: An introduction to copula. Springer, New York (1998)
4. Cherubini, U., Luciano, E., Vecchiato, W.: Copula Methods in Finance. Wiley (2004)
5. González-Fernández, Y., Soto, M.: copulaedas: An R Package for Estimation of Distribution Algorithms Based on Copulas (2013), Preprint arXiv:  
<http://arxiv.org/abs/1209.5429>
6. Póczos, B., Ghahramani, Z., Schneider, J.: Copula-based Kernel dependency measures. In: Proceedings of the International Conference on Machine Learning, ICML 2012 (2012)
7. Joe, H.: Families of m-variate distributions with given margins and  $m(m-1)/2$  bivariate dependence parameters. In: Rüschendorf, L., Schweizer, B., Taylor, M.D. (eds.) Distributions with Fixed Marginals and Related Topics, pp. 120–141 (1996)
8. Cuesta-Infante, A., Santana, R., Hidalgo, J.I., Bielza, C., Larrañaga, P.: Bivariate empirical and n-variate archimedean copulas in estimation of distribution algorithms. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, pp. 1355–1362 (2010)
9. Joe, H.: Multivariate Models and Dependence Concepts. Chapman & Hall (1997)
10. Genest, C., Rivest, L.-P.: Statistical inference procedures for bivariate Archimedean copulas. J. Am. Stat. Assoc. 88(423), 1034–1043 (1993)
11. Guo, X., Wang, L., Zeng, J., Zhang, X.: VQ Codebook Design Algorithm Based on Copula Estimation of Distribution Algorithm. In: First International Conference on Robot, Vision and Signal Processing, pp. 178–181 (2011)
12. DelaOssa, L., Gámez, J.A., Mateo, J.L., Puerta, J.M.: Avoiding premature convergence in estimation of distribution algorithms. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009, pp. 455–462 (2009)