Using Copulas in Estimation of Distribution Algorithms

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R. Villa-Diharce

Centro de Investigación en Matemáticas, Guanajuato, México
{rsalinas,artha,villadi}@cimat.mx

Abstract. A new way of modeling probabilistic dependencies in Estimation of Distribution Algorithm (EDAs) is presented. By means of copulas it is possible to separate the structure of dependence from marginal distributions in a joint distribution. The use of copulas as a mechanism for modeling joint distributions and its application to EDAs is illustrated on several benchmark examples.

1 Introduction

Estimation of Distribution Algorithms (EDAs) [17] are recognized as a new paradigm in Evolutionary Computation to deal with optimization problems [15]. EDAs are a class of Evolutionary Algorithms (EAs) based on probabilistic models instead of genetic operators such as crossover and mutation. The use of probabilistic models allow us to explicitly represent: 1) dependencies between the decision variables; and 2) their structure. EDAs populate the next generation by simulating individuals from the probabilistic model, therefore, the goal is to transfer both the data dependencies and the structure found in the best individuals into the next population. A pseudocode for EDAs is shown in Algorithm 1.

Algorithm 1. Pseudocode for EDAs

1: assign $t \longleftarrow 0$
   generate the initial population P_0 with M individuals at random
2: select a collection of N solutions S_t, with $N < M$, from P_t
3: estimate a probabilistic model $\mathcal{M}_t$ from S_t
4: generate the new population by sampling from the distribution of S_t.
   assign $t \longleftarrow t + 1$
5: if stopping criterion is not reached go to step 2

As it can be seen in step 3, the interactions among the decision variables are taken into account through the estimated model. The possibility of incorporating the dependencies among the variables into the new population greatly modifies the performance of an EDA. Nowadays, several EDAs have been proposed for

A. Hernández Aguirre et al. (Eds.): MICAI 2009, LNAI 5845, pp. 658–668, 2009.

optimization problems in discrete and continuous domains. They can be grouped by the complexity of the probabilistic model used to learn the interactions between the variables. For instance, the UMDA [18,11,13] can be considered the most simple EDA because it does not take into account dependencies between the variables, while the BMDA [21] and MIMIC [6,11,13] just take dependencies between pairs of variables into account. Probabilistic models such as Bayesian networks and multivariate Gaussian distributions have been used by EDAs for multiple dependencies. Some examples in discrete domain are PADA [24], EBNA [8,12] and BOA [20]. For continuous domain EMNA [14] and EGNA [11,13] are EDAs based on multivariate and bivariate Gaussian distribution respectively. In this paper we deal with continuous optimization problems, and address the use of copulas in EDAs. One motivation of this work is to take advantage of the almost natural capacity of copulas to represent bivariate dependencies through concordance measures, such as Kendall's tau or Pearson's rho. The goal of the paper is to introduce copula functions and implement the copula-based MIMIC; to the best of our knowledge this paper would be one of the first studies on the performance of EDAs based on copulas. Related works have considered EDAs based on multidimensional Gaussian copula with nonparametric marginals [1,3] and EDAs based on two dimensional copulas with Gaussian marginals [27,26].

The structure of the paper is the following: Section 2 is a short introduction to copula functions, Sect. 3 describes the implementation of the MIMIC algorithm with copula functions. Section 4 presents the experimental setting to solve 5 test global optimization problems, and Sect. 5 resumes the conclusions.

2 Copula Functions

The concept of copula was introduced by Sklar [23] to separate the effect of dependence from the effect of marginal distributions in a joint distribution. The separation between marginal distributions and a dependence structure explains the modeling flexibility given by copulas and, for this reason, they have been widely used in many research and application areas such as finance [4,25], climate [22], oceanography [7], hydrology [9], geodesy [2], and reliability [16].

Definition 1. *A copula is a joint distribution function of standard uniform random variables. That is,*

$$C(u_1, \ldots, u_n) = Pr[U_1 \leq u_1, \ldots, U_n \leq u_n] \ ,$$

where $U_i \sim U(0,1)$ *for* $i = 1, \ldots, n$.

For a more formal definition of copulas, the reader is referred to [10,19]. The following result, known as Sklar's theorem, connects marginal distributions and copula with a joint distribution.

Theorem 1 (Sklar). *Let F be a n-dimensional distribution function with marginals $F_1, F_2, \ldots, F_n$, then there exists a copula C such that for all x in $\overline{\mathbb{R}}^n$,*

$$F(x_1, x_2, \ldots, x_n) = C(F_1(x_1), F_2(x_2), \ldots, F_n(x_n)) \ ,$$

where $\overline{\mathbb{R}}$ denotes the extended real line $[-\infty, \infty]$. If $F_1(x_1), F_2(x_2), \ldots, F_n(x_n)$ are all continuous, then C is unique. Otherwise, C is uniquely determined on $Ran(F_1) \times Ran(F_2) \times \cdots Ran(F_n)$, where Ran stands for the range.

According to Theorem 1, the n-dimensional density f can be represented as

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) \cdot f_2(x_2) \cdots f_n(x_n) \cdot c(F_1(x_1), F_2(x_2), \ldots, F_n(x_n)) \ ,$$

where $f_i(x_i)$ is the density of variable x_i and c is the density of the copula C. This result allows us to choose different marginals and a dependence structure given by the copula and then merge them to build a multivariate distribution. This contrasts with the usual way to construct multivariate distributions, which suffers from the restriction that the margins are usually of the same type.

There are many families of copulas and each of them is characterized by a parameter or a vector of parameters. These parameters measure dependence between the marginals and are called *dependence parameters* $\boldsymbol{\theta}$. In this paper we use bivariate copulas with one dependence parameter θ. The dependence parameter is related to Kendall's tau through the equation (see [19])

$$\tau(X_1, X_2) = 4 \int_0^1 \int_0^1 C(u_1, u_2; \theta) dC(u_1, u_2; \theta) - 1 \ . \tag{1}$$

Kendall's tau measures the concordance between two continuous random variables X_1 and X_2. Table 1 shows the defining equations of the Frank copula and the Gaussian copula. Observe how the dependence parameter in the copula function is related to the Kendall's tau. The dependence parameter of a bivariate copula can be estimated using the maximum likelihood method. To do so, we need to maximize the *log-likelihood* function given by

$$l(\theta) = \sum_{t=1}^{T} \ln\, c(F(x_{1t}), F(x_{2t}); \theta) \ ,$$

where T is the sample size. The value θ which maximizes the log-likelihood is called *maximum likelihood estimator* $\widehat{\theta}_{\mathrm{MLE}}$. Once the value of θ is estimated, the bivariate copula is well defined. For maximizing the likelihood function we use the nonparametric estimation of θ given by Kendall's tau in (1) as an initial approximation to $\widehat{\theta}_{\mathrm{MLE}}$.

3 An EDA Based on Copula Functions

In order to show how a probabilistic model based on copulas can be used in EDAs we proposed an adaptation of the $MIMIC_C^G$ [11,13] with no Gaussian

Table 1. Bivariate copulas used in this paper

Copula's name	Description	
Frank	Distribution:	$C(u_1,u_2;\theta) = -\frac{1}{\theta}\ln\left(1+\frac{(e^{-\theta u_1}-1)(e^{-\theta u_2}-1)}{e^{-\theta}-1}\right)$
	Parameter:	$\theta \in (-\infty,\infty)$
	Kendall's tau:	$\tau = 1-\frac{4}{\theta}[1-D_1(\theta)]$, where $D_1(\theta)=\frac{1}{\theta}\int_0^\theta \frac{t}{e^t-1}dt$
Gaussian	Distribution:	$C(u_1,u_2;\theta) = \Phi_{\mathrm{G}}\left(\Phi^{-1}(u_1),\Phi^{-1}(u_2)\right)$, where Φ_{G} is the standard bivariate normal distribution with correlation parameter θ
	Parameter:	$\theta \in (-1,1)$
	Kendall's tau:	$\tau = \frac{2}{\pi}\arcsin(\theta)$
	Entropy:	$H(U_1,U_2) = \frac{1}{2}\log(1-\theta^2)$

assumption over univariate and bivariate density functions. Next, for completeness sake, we describe the principles of the $MIMIC_C^G$ learning algorithm.

Given a permutation of the numbers between 1 and n, $\pi = (i_1, i_2, \ldots, i_n)$ we define a class of density functions, $f_\pi(\boldsymbol{x})$:

$$f_\pi(\boldsymbol{x}) = f(x_{i_1}|x_{i_2}) \cdot f(x_{i_2}|x_{i_3}) \cdots f(x_{i_{n-1}}|x_{i_n}) \cdot f(x_{i_n}) \ . \tag{2}$$

Our goal is to choose the permutation π that minimizes the Kullback-Leibler divergence between the true density function $f(\boldsymbol{x})$ and the proposed density function $f_\pi(\boldsymbol{x})$:

$$D_{KL}\left(f(\boldsymbol{x})||f_\pi(\boldsymbol{x})\right) = E_{f(\boldsymbol{x})}\left[\log\frac{f(\boldsymbol{x})}{f_\pi(\boldsymbol{x})}\right] \ .$$

It is well known that conditional entropy $H(X|Y)$ and mutual information $I(X,Y)$ are related in the following way:

$$H(X|Y) = -I(X,Y) + H(X) \ ,$$

where $H(X) = -E_{f(x)}[\log f(x)]$ denotes the entropy of the continuous random variable X with density $f(x)$. The Kullback-Liebler divergence can be written as:

$$\begin{aligned} D_{KL}\left(f(\boldsymbol{x})||f_{\pi}(\boldsymbol{x})\right) &= -H(\boldsymbol{X}) + \sum_{k=1}^{n-1} H(X_{i_k}|X_{i_{k+1}}) + H(X_{i_n}) \\ &= -H(\boldsymbol{X}) + \sum_{k=1}^{n} H(X_{i_k}) - \sum_{k=1}^{n-1} I(X_{i_k}, X_{i_{k+1}}) \ . \end{aligned}$$

The first two terms in the divergence do not depend on π. Therefore, minimize the Kullback-Leibler is equivalent to maximize

$$J_{\pi}(\boldsymbol{X}) = \sum_{k=1}^{n-1} I(X_{i_k}, X_{i_{k+1}}) \ ,$$

where

$$I(X_{i_k}, X_{i_{k+1}}) = E_{f(x_{i_k}, x_{i_{k+1}})}\left[\log \frac{f(x_{i_k}, x_{i_{k+1}})}{f(x_{i_k}) \cdot f(x_{i_{k+1}})}\right] \ .$$

According to [6], the optimal permutation π is the one that equivalently produces the highest pairwise mutual information with respect to the true distribution. But due to computational efficiency reasons we will employ the greedy algorithm originally proposed by [6] and adapted by [11]. Thus, the $MIMIC$ learning algorithm is based on a dependence test, and this is measured through mutual information. In this paper we will use the following fact (see [5]) between copula entropy and mutual information:

$$\begin{aligned} -E_{c(u_1,u_2)}[\log c(u_1, u_2)] &= -E_{f(x_1,x_2)}\left[\log \frac{f(x_1, x_2)}{f(x_1) \cdot f(x_2)}\right] \\ H(U_1, U_2) &= -I(X_1, X_2) \ , \end{aligned}$$

where $U_1 = F(X_1)$ and $U_2 = F(X_2)$.

Our proposed EDA uses two different dependence functions: a Frank copula and a Gaussian copula. These copulas are chosen because their dependence parameter have associated all range values of Kendall's tau. This means that negative and positive dependence between the marginals are considered in both copulas. However, they differ in the way they model extreme and centered values [25]. For instance, a Frank copula is mostly appropiate for data that exhibit weak dependence between extreme values and strong dependence between centered values. The proposed EDA estimates the copula entropy between each pair of variables in order to calculate the mutual information. The pair of variables with the largest mutual information are selected as the two first variables of the permutation π. The following variables of π are chosen according to their mutual information with respect to the previous variable. Algorithm 2 shows a straightforward greedy algorithm to find a permutation π.

Algorithm 2. Greedy algorithm to pick a permutation π

1: choose $(i_n, i_{n-1}) = \arg\max_{j \neq k} \widehat{I}(X_j, X_k)$, where $\widehat{I}()$ is an estimation of the mutual information between two variables.
2: choose $i_k = \arg\max_j \widehat{I}(X_{i_{k+1}}, X_j)$, where $j \neq i_{k+1}, \ldots, i_n$ and $k = n-1, n-2, \ldots, 2, 1$.

For a Gaussian copula there is a direct way to calculate its entropy and mutual information; for a Frank copula we estimate its entropy with a numerical approximation.

Once a permutation π is found, generating samples follows the order established by (2). In order to do it, we first sample variable $U_{i_n} \sim U(0,1)$ and then we sample variables $U_{i_k} \sim C(U_{i_k} | U_{i_{k+1}} = u_{i_{k+1}})$ from conditional copula of U_{i_k} given the value of $U_{i_{k+1}}$ for $k = n-1, \ldots, 1$. After that, we use values of U_i to find quantiles X_i through expression $X_i = F_{X_i}^{-1}(U_i)$.

It is important to say that, by means of copulas, we can write (2) as

$$f_\pi(\boldsymbol{x}) = \prod_{i=1}^{n} f(x_i) \cdot \prod_{k=1}^{n-1} c(u_{i_k}, u_{i_{k+1}}) \ . \tag{3}$$

where $u_{i_k} = F(x_{i_k})$ and $u_{i_{k+1}} = F(x_{i_{k+1}})$. This means that $MIMIC_C^G$ is a particular copula based EDA with Gaussian copulas $C(u_{i_k}, u_{i_{k+1}})$ and Gaussian marginals $F(x_i)$, for $k = n-1, \ldots, 1$ and $i = 1, \ldots, n$.

In this work we use Beta distributions as marginals. In order to estimate the parameters of the probabilistic model (3), we use the Inference Function for Margins method (IFM) [4]. This method is based on maximum likelihood and estimates first the parameters of marginals and then use them to estimate parameters of copulas. The test problems used in this paper have bounded search space. Each value of variable X_i from search space is transformed to a value in $(0,1)$ through a linear transformation. This explains why we use Beta distributions as marginals.

We summarize in Algorithm 3 the proposed approach. The main aspects, such as the estimation of the probabilistic model and the generation of the new population, are shown.

4 Experiments

We use three algorithms in order to optimize five test problems. The algorithms are $MIMIC_C^G$, copula based EDA using Frank copulas, and copula based EDA using Gaussian copulas. Table 2 shows the definition of the test problems used in the experiments: Ackley, Griewangk, Rastrigin, Rosenbrock, and Sphere functions. We use test problems in 10 dimensions. Each algorithm is run 30 times for each problem. The population size is 100. The maximum number of evaluations is 300,000. However, when convergence to a local minimum is detected the run is stopped. Any improvement less than 1×10^{-6} in 25 iterations is considered convergence. The goal is to reach the optimum with an error less than 1×10^{-6}.

Algorithm 3. Pseudocode for estimating model and generating new population

1: calculate pairwise mutual information using copula entropy
2: use greedy algorithm to pick a permutation (Algorithm 2)
3: calculate concordance measure Kendall's tau between variables in permutation π
4: obtain an initial approximation to the dependence parameter θ_τ using relationship with Kendall's tau (Table 1)
5: estimate marginal and copula parameters using Inference Function for Margins Method with θ_τ as initial approximation
6: simulate U_{i_n} from uniform distribution $U(0,1)$
7: simulate U_{i_k} from conditional copula $C(U_{i_k}|U_{i_{k+1}})$, $k = n-1, n-2, \ldots, 2, 1$
8: determine X_i using quasi-inverse $F^{-1}_{X_i}(U_i)$, $i = 1, \ldots, n$

Table 2. Test functions

Name		Description
Ackley	Function:	$F(\boldsymbol{x}) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n} \cdot \sum_{i=1}^n x_i^2}\right)$ $-\exp\left(\frac{1}{n} \cdot \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \exp(1)$
	Search space:	$-10 \leq x_i \leq 10$, $i = 1, \ldots, 10$
	Minimum value:	$F(\mathbf{0}) = 0$
Griewangk	Function:	$F(\boldsymbol{x}) = 1 + \sum_{i=1}^n \frac{x_1^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$
	Search space:	$-600 \leq x_i \leq 600$, $i = 1, \ldots, 10$
	Minimum value:	$F(\mathbf{0}) = 0$
Rastrigin	Function:	$F(\boldsymbol{x}) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$
	Search space:	$-5.12 \leq x_i \leq 5.12$, $i = 1, \ldots, 10$
	Minimum value:	$F(\mathbf{0}) = 0$
Rosenbrock	Function:	$F(\boldsymbol{x}) = \sum_{i=1}^{n-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$
	Search space:	$-10 \leq x_i \leq 10$, $i = 1, \ldots, 10$
	Minimum value:	$F(\mathbf{1}) = 0$
Sphere model	Function:	$F(\boldsymbol{x}) = \sum_{i=1}^n x_i^2$
	Search space:	$-600 \leq x_i \leq 600$, $i = 1, \ldots, 10$
	Minimum value:	$F(\mathbf{0}) = 0$

4.1 Numerical Results

In Table 3 we report the fitness value reached by the algorithms in all test functions. The information about the number of evaluations required by each algorithm is reported in Table 4.

To properly compare the performance of the algorithms (using the optimum value reached), we conducted a hypothesis test based on a Bootstrap method for the differences between the means of the three comparison pairs, for all test problems. Table 5 shows the confidence interval for the means, and the corresponding p-value.

Table 3. Descriptive fitness results for all test functions

Algorithm	Best	Median	Mean	Worst	Std. deviation
Ackley					
$MIMIC_C^G$	6.47E-007	8.65E-007	8.62E-007	9.97E-007	1.06E-007
Frank copula	5.79E-007	2.29E-006	3.06E-003	4.71E-002	9.31E-003
Gaussian copula	5.62E-007	9.07E-007	3.64E-006	7.80E-005	1.41E-005
Griewangk					
$MIMIC_C^G$	3.92E-007	8.66E-007	1.30E-003	3.88E-002	7.09E-003
Frank copula	4.30E-007	9.38E-007	2.99E-003	2.90E-002	6.81E-003
Gaussian copula	1.46E-007	8.11E-007	1.81E-002	4.31E-001	7.85E-002
Rastrigin					
$MIMIC_C^G$	4.17E-007	9.96E-001	3.37E+000	2.33E+001	6.24E+000
Frank copula	2.21E+000	4.99E+000	8.05E+000	3.69E+001	9.43E+000
Gaussian copula	7.49E-007	4.00E+000	5.48E+000	2.68E+001	5.35E+000
Rosenbrock					
$MIMIC_C^G$	7.31E+000	8.03E+000	8.89E+000	2.43E+001	3.17E+000
Frank copula	6.87E+000	7.83E+000	7.95E+000	9.69E+000	6.44E-001
Gaussian copula	6.26E+000	8.15E+000	8.53E+000	1.48E+001	1.78E+000
Sphere					
$MIMIC_C^G$	3.55E-007	7.00E-007	7.10E-007	9.86E-007	2.02E-007
Frank copula	3.39E-007	7.40E-007	3.03E-001	8.23E+000	1.50E+000
Gaussian copula	3.42E-007	8.92E-007	4.85E-001	1.22E+001	2.23E+000

Table 4. Descriptive function evaluations for all test functions

Algorithm	Mean	Std. deviation
Ackley		
$MIMIC_C^G$	7660.30	131.24
Frank copula	9310.30	1761.88
Gaussian copula	7657.00	675.63
Griewangk		
$MIMIC_C^G$	6927.70	1581.97
Frank copula	8343.40	2460.13
Gaussian copula	7835.20	2825.34
Rastrigin		
$MIMIC_C^G$	11788.60	3146.69
Frank copula	17055.40	5262.08
Gaussian copula	15408.70	4511.01
Rosenbrock		
$MIMIC_C^G$	12841.30	2665.61
Frank copula	14280.10	1355.85
Gaussian copula	14016.10	1666.29
Sphere		
$MIMIC_C^G$	6175.30	154.87
Frank copula	7069.60	2829.51
Gaussian copula	7874.80	3144.74

Table 5. Results for the difference between fitness means in each problem. A 95% interval confidence and a p-value are obtained through a Bootstrap technique.

Compared algorithms	95% Interval		p-value
Ackley			
$MIMIC_C^G$ vs. Frank copula	-6.15E-03	-7.37E-04	8.13E-02
$MIMIC_C^G$ vs. Gaussian copula	-7.89E-06	1.69E-08	1.94E-01
Frank copula vs. Gaussian copula	7.28E-04	6.14E-03	8.17E-02
Griewangk			
$MIMIC_C^G$ vs. Frank copula	-4.47E-03	1.29E-03	3.26E-01
$MIMIC_C^G$ vs. Gaussian copula	-4.50E-02	-4.33E-04	1.62E-01
Frank copula vs. Gaussian copula	-4.34E-02	1.37E-03	2.60E-01
Rastrigin			
$MIMIC_C^G$ vs. Frank copula	-8.11E+00	-1.48E+00	2.48E-02
$MIMIC_C^G$ vs. Gaussian copula	-4.49E+00	3.20E-01	1.60E-01
Frank copula vs. Gaussian copula	-5.09E-01	5.87E+00	1.89E-01
Rosenbrock			
$MIMIC_C^G$ vs. Frank copula	1.34E-01	2.01E+00	1.12E-01
$MIMIC_C^G$ vs. Gaussian copula	-6.13E-01	1.50E+00	5.68E-01
Frank copula vs. Gaussian copula	-1.18E+00	-6.44E-02	9.48E-02
Sphere			
$MIMIC_C^G$ vs. Frank copula	-8.51E-01	-1.16E-03	1.45E-01
$MIMIC_C^G$ vs. Gaussian copula	-1.28E+00	-1.72E-02	1.42E-01
Frank copula vs. Gaussian copula	-9.84E-01	5.46E-01	6.80E-01

4.2 Discussion

For the Ackley problem, intervals confidence show signicant differences between $MIMIC_C^G$ and Gaussian copula against Frank copula. This means that a dependence structure based on Gaussian copula is more adequate than a dependence structure based on Frank copula.

For the Griewangk problem, the algorithm that shows the best behaviour is $MIMIC_C^G$, closely followed by Frank copula algorithm. In this case, interval confidence between $MIMIC_C^G$ and Gaussian copula shows that better results are obtained using both Gaussian dependence structure and marginals.

The $MIMIC_C^G$ is the algorithm that performed best for the Rastrigin problem. For this problem there is statistically significant difference in the mean fitness between the $MIMIC_C^G$ and the Frank copula algorithms. Although results of Frank copula algorithm are not statistically different of Gaussian copula, is more suitable for this problem to choose a Gaussian structure than a Frank dependence. Respect to marginals distributions we can say something similar between Gaussian copula algorithm and $MIMIC_C^G$ in the sense that is more adequate to choose Gaussian marginals than Beta marginals.

For the Rosenbrock problem, the intervals confidence shows statistical differences between $MIMIC_C^G$ and Gaussian copula against Frank copula. In this case, a Frank dependence between marginals is more adecuate than a Gaussian strucuture between marginals. Fitness results between $MIMIC_C^G$ and Gaussian copula algorithm show no difference between Gaussian or Beta marginals if structure dependence is modeled by a Gaussian copula.

Finally, the fitness results for Sphere problem indicate that $MIMIC_C^G$ obtained the global minimum in all the executions. The selection of Gaussian structure and Gaussian marginals is more adecuate for this problem.

Regarding the number of fitness function evaluations, Table 4, the three algorithms performed in a similar way.

5 Conclusions

In this paper we introduce the use of copulas in EDAs. According to numerical experiments the selection of a copula for modeling structure dependence and the selection of marginals distributions can help achieving better fitness results. Although we use the same structure dependence and the same marginals for each algorithm, it is not necessary to do it. Fitness results are the result of the selected structure dependences and marginals.

The three algorithms performed very similar, however, more experiments are necessary with different probabilistic models in order to identify where the copula functions mean a clear advantage to EDAs.

References

1. Arderí-García, R.J.: Algoritmo con Estimación de Distribuciones con Cópula Gaussiana. Bachelor's thesis, Universidad de La Habana. La Habana, Cuba (2007) (in Spanish)
2. Bacigál, T., Komorníková, M.: Fitting Archimedean copulas to bivariate geodetic data. In: Rizzi, A., Vichi, M. (eds.) Compstat 2006 Proceedings in Computational Statistics, pp. 649–656. Physica-Verlag HD, Heidelberg (2006)
3. Barba-Moreno, S.E.: Una propuesta para EDAs no paramétricos. Master's thesis, Centro de Investigación en Matemáticas. Guanajuato, México (2007) (in Spanish)
4. Cherubini, U., Luciano, E., Vecchiato, W.: Copula Methods in Finance. Wiley, Chichester (2004)
5. Davy, M., Doucet, A.: Copulas: a new insight into positive time-frequency distributions. Signal Processing Letters, IEEE 10(7), 215–218 (2005)
6. De Bonet, J.S., Isbell, C.L., Viola, P.: MIMIC: Finding Optima by Estimating Probability Densities. In: Advances in Neural Information Processing Systems, vol. 9, pp. 424–430. The MIT Press, Cambridge (1997)
7. De-Waal, D.J., Van-Gelder, P.H.A.J.M.: Modelling of extreme wave heights and periods through copulas. Extremes 8(4), 345–356 (2005)
8. Etxeberria, R., Larrañaga, P.: Global optimization with Bayesian networks. In: Ochoa, A., Soto, M., Santana, R. (eds.) Second International Symposium on Artificial Intelligence, Adaptive Systems, CIMAF 1999, Academia, La Habana, pp. 332–339 (1999)
9. Genest, C., Favre, A.C.: Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask. Journal of Hydrologic Engineering 12(4), 347–368 (2007)
10. Joe, H.: Multivariate models and dependence concepts. Chapman and Hall, London (1997)

11. Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Optimization by learning and simulation of Bayesian and Gaussian networks. Technical report KZZA-IK-4-99. Department of Computer Science and Artificial Intelligence, University of the Basque Country (1999)
12. Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Combinatorial optimization by learning and simulation of Bayesian networks. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 343–352 (2000)
13. Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: Wu, A.S. (ed.) Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, pp. 201–204 (2000)
14. Larrañaga, P., Lozano, J.A., Bengoetxea, E.: Estimation of Distribution Algorithm based on multivariate normal and Gaussian networks. Technical report KZZA-IK-1-01. Department of Computer Science and Artificial Intelligence, University of the Basque Country (2001)
15. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
16. Monjardin, P.E.: Análisis de dependencia en tiempo de falla. Master's thesis, Centro de Investigación en Matemáticas. Guanajuato, México (2007) (in Spanish)
17. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions I. Binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
18. Mühlenbein, H.: The Equation for Response to Selection and its Use for Prediction. Evolutionary Computation 5(3), 303–346 (1998)
19. Nelsen, R.B.: An Introduction to Copulas. Springer, Heidelberg (2006)
20. Pelikan, M., Goldberg, D.E., Cantú-Paz, E.: BOA: The Bayesian optimization algorithm. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference GECCO 1999, vol. 1, pp. 525–532. Morgan Kaufmann Publishers, San Francisco (1999)
21. Pelikan, M., Mühlenbein, H.: The Bivariate Marginal Distribution Algorithm. In: Roy, R., Furuhashi, T., Chawdhry, P.K. (eds.) Advances in Soft Computing - Engineering Design and Manufacturing, pp. 521–535. Springer, Heidelberg (1999)
22. Schölzel, C., Friederichs, P.: Multivariate non-normally distributed random variables in climate research – introduction to the copula approach. Nonlinear Processes in Geophysics 15(5), 761–772 (2008)
23. Sklar, A.: Fonctions de répartition à n dimensions et leurs marges. Publications de l'Institut de Statistique de l'Université de Paris 8, 229–231 (1959)
24. Soto, M., Ochoa, A., Acid, S., de Campos, L.M.: Introducing the polytree approximation of distribution algorithm. In: Ochoa, A., Soto, M., Santana, R. (eds.) Second International Symposium on Artificial Intelligence, Adaptive Systems, CIMAF 1999, Academia, La Habana, pp. 360–367 (1999)
25. Trivedi, P.K., Zimmer, D.M.: Copula Modeling: An Introduction for Practitioners. In: vol.1 of Foundations and Trends® in Econometrics Now Publishers (2007)
26. Wang, L.F., Zeng, J.C., Hong, Y.: Estimation of Distribution Algorithm Based on Archimedean Copulas. In: GEC 2009: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 993–996. ACM, New York (2009)
27. Wang, L.F., Zeng, J.C., Hong, Y.: Estimation of Distribution Algorithm Based on Copula Theory. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1057–1063. IEEE Press, Los Alamitos (2009)