

A Multi-population Parallel Estimation of Distribution Algorithms Based on Clayton and Gumbel Copulas

Chunyan Chang¹ and Lifang Wang²

¹ College of Textile Engineering and Art, Taiyuan University of Technology,
Yuci, Jingzhong, 030600, P.R. China
sxccya@163.com

² Complex System and Computational Intelligence Laboratory, Taiyuan University of Science
and Technology, Taiyuan, 030024, P.R. China
wlf1001@163.com

Abstract. The idea of multi-population parallel strategy and the copula theory are introduced into the Estimation of Distribution Algorithm (EDA), and a new parallel EDA is proposed in this paper. In this algorithm, the population is divided into some subpopulations. Different copula is used to estimate the distribution model in each subpopulation. Two copulas, Clayton and Gumbel, are used in this paper. To estimate the distribution function is to estimate the copula and the margins. New individuals are generated according to the copula and the margins. In order to increase the diversity of the subpopulation, the elites of one subpopulation are learned by the other subpopulation. The experiments show the proposed algorithm performs better than the basic copula EDA and some classical EDAs in speed and in precision.

Keywords: multi-population, parallel, estimation of distribution algorithms (EDAs), copula theory, Clayton, Gumbel.

1 Introduction

The Estimate of Distribution Algorithms (EDAs) are new Evolutionary Algorithms (EAs) combined Genetic Algorithms (GAs) with statistical learning. EDAs realize population evolution by iteratively estimating probabilistic model of selected population which describes the solutions distribution from a macroscopic perspective, and then sampling the new individuals from the model. The probabilistic model reflect the relationship between variables, therefore EDAs are effectively used to obtain optimized solutions for complex problems [1]. Currently, many kinds of EDAs were studied by scholars. The classical algorithms are: (1) PBIL [2] and UMDA [3], in which the variables are independent; (2) MIMIC[4] and BMDE[5], which only consider the dependences between pairs of variables; (3) ECGA[6], FDA[7] and BOA[8], which accept the multivariate correlation; (4) UMDAc[9], PBILc[10] and EMNA [11], which focus on EDAs in continuous search spaces, etc. Estimating probabilistic model is the key step in EDAs, with the problem and model become

more and more complex, the consumption of algorithm in runtime and storage space will be increased.

Copula theory becomes the hotspot in the areas of statistics and finance. In this theory, the joint probability distribution function is divided into their one-dimensional marginal distribution functions and a copula function. And the joint probability distribution function is an effectual tool to mirror the relationship between variables. Therefore, the complexity of EDAs and operation time of modeling and sampling will decrease by using of copula. Some authors present new EDAs based on copula theory (copula EDAs), and make some incipient activities [12], [13]. The main idea of copula EDAs is that firstly, select dominant population, and then modeling the marginal distribution functions from them and sampling from a copula function.

The copula function plays an important role in copula EDAs. There are two classes of copulas to express the multivariate correlations: Elliptical copulas and Archimedean copulas. Elliptical copulas include Gauss-copula and t-copula. Archimedean copulas are produced from different generators according to the definition of Archimedean copula [14]. About 200 kinds of Archimedean copulas are currently being developed, in which Clayton, Gumbel and Frank are famous and in regular use. Every copula has special characteristic. Existing copula EDAs estimate the dependence structure of variables using an pre-assigned copula function, in which neither discuss the population whether exactly obeys the distribution modeled by this copula nor measure this copula function whether can fully describe the dependence structure of variables.

To deal with early convergence of traditional EDAs and single representation of relationship between variables in copula EDAs, we propose a multi-population parallel EDA based on copula theory in this paper. This algorithm optimizes more quickly and effectively than traditional EDAs and copula EDAs.

2 A Brief Introduction of Parallel Genetic Algorithms

The parallel Genetic Algorithms (pGAs) are the improved algorithms for traditional GAs. They either have inherently parallelism of GAs or have high-speed performance of parallel computer, so the global searching ability of the pGAs is improved and premature convergence is overcome available.

There are three main types of pGAs: (1) global single-population master-slave GAs, (2) single-population fine-grained GAs, (3) multiple-population coarse-grained GAs [15]. We are going to focus on the coarse-grained model because it has stronger suitability and wider utility. According to the coarse model, the overall population is distributed over multiple subpopulations and occasionally allows the migration or exchange of some individuals among the different subpopulations. Therefore, each subpopulation executes an independent algorithm. The migration strategy of coarse-grained contains some parameters [16]: (1) How often (in generations) is information sent? It is configured with migration frequency. (2) How many individuals migrate each time? It is configured with migration rate. (3) What information is selected to migrate? It is configured with information selection. (4) How are the incoming information and the local algorithm state combined? It is configured with acceptance policy. (5) Which island sends information to which other? It is configured with migration topology.

3 Contribution

In this paper, a new copula EDA named multi-population parallel copula EDA is presented.

3.1 Multi-population Parallel Copula EDA

According to the framework of copula EDA [12], the copula EDA is an iterative run containing three main steps. The first step is to select a subpopulation denoted by $x=\{x^j=(x_1^j,x_2^j,\dots,x_n^j), j=1,2,\dots,s\}$. The second step is to estimate the margins F_i for each random variable $X_i, i=1,2,\dots,n$ according to the samples $\{x_i^j, j=1,2,\dots,s\}$. The last step is to select or to construct a copula C according to x , and then sample from C . Assuming the sampled vectors are $\{u^{(k)}=(u_1^{(k)},u_2^{(k)},\dots,u_n^{(k)}), k=1,2,\dots,l\}$, the new individuals $\{x^{(k)}=(x_1^{(k)},x_2^{(k)},\dots,x_n^{(k)}), k=1,2,\dots,l\}$ can be calculate3d by using

$$x_i^{(k)} = F_i^{-1} \left(u_i^{(k)} \right), i = 1, \dots, n, k = 1, \dots, l \tag{1}$$

where, F_i^{-1} is the inverse function of the i th marginal distribution function.

copula EDA replaces some old individuals of the original population with the new generated individuals and progresses the new evolution until the terminate condition is met.

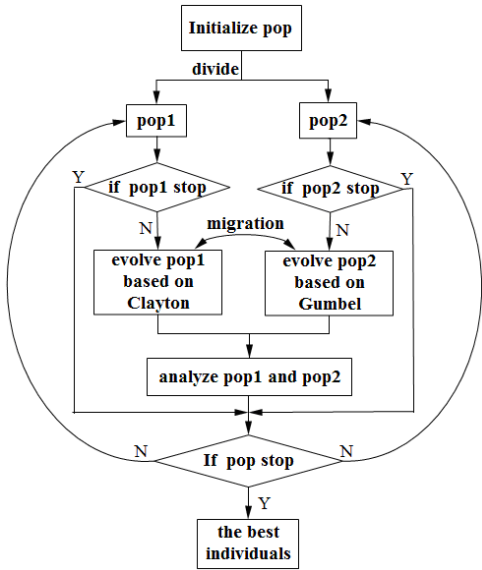


Fig. 1. The flow chart of multi-population parallel copula EDA

Some authors have studied on the optimization effectiveness of different copula functions under the framework of copula EDA [17],[18]. The results show that in many cases, the different copulas are used in common basic test function to bring about

different performances, because each copula function has special characteristic. So modeling and sampling by a pre-assigned single copula function without judgment is difficult to exactly describe the multivariate correlation of practical problem.

In order to possibly speed up the search of EDAs and to increase the diversity of the population, the idea of parallel Genetic Algorithms is introduced into copula EDA. Multi-population parallel copula EDA separates the individuals into certain independent subpopulations. According to the frame work of copula EDA, each subpopulation use different copulas to model and sample, respectively. We choose two famous Archimedean copulas: Clayton and Gumbel. The sampling algorithm from copula is the algorithm proposed by Marshall and Olkin[19] , [20].

Concluding the analysis presented above, the multi-population parallel copula EDA works as Figure 1 and Algorithm 1, where, the procedure Clatyton sample and Gumbel sample is shown in Algorithm 2 and Algorithm 3 respectively.

Algorithm 1. Pseudo code for multi-population parallel copula EDA.

Step1. Randomly generate initial population with size N , and divide them into two subpopulations: $pop1_0$, $pop2_0$. and set $g \leftarrow 0$.

Step2. Initialize evaluation flag: $evFlag1 \leftarrow 1$ $evFlag2 \leftarrow 1$.

Step3. Select two subpopulations $pop1_g$ and $pop2_g$ with same size of s from $pop1_g$, $pop2_g$ According to certain select-strategy, respectively.

Step4. If $evFlag1 = 1$, evolve $pop1_g$ based on Clayton copula.

Step4-1 Estimate the univariate marginal distribution function \hat{F}_i for each dimension from $pop1_g$.

Step4-2. For $k=1$ to l do $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) = \text{Clayton_sample}(k)$.

Step4-3. Replace the old individuals of $pop1_g$ with the new individual.

Step4-4. If the stopping criterion of $pop1_g$ is reached, set $evFlag1 \leftarrow 0$.

Step5. If $evFlag2 = 1$, evolve, $pop2_g$ base on Gumbel copula.

Step5-1. Estimate the univariate marginal distribution function F_i for each dimension from $pop2_g$.

Step5-2. For $k=1$ to l do $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) = \text{Gumbel_sample}(k)$.

Step5-3. Replace the old individuals of $pop2_g$ with the new individuals.

Step5-4. If evolutionary stopping criterion of $pop2_g$ is reached, set $evFlag2 \leftarrow 0$.

Step6. If $(g \% c == 0)$ $pop1_g$ and $pop2_g$ mutually migration w best individuals.

Step7. Set $g \leftarrow g + 1$.

Step8. If stopping criterion is met, then stop the algorithm and the best individual is the optimization result, else go to *Step3*.

3.2 Clayton Copula Model Sampling

The generator of Clayton copula is

$$\varphi(t) = (t^{-\theta} - 1) / \theta \quad (2)$$

the inverse function is

$$\varphi^{-1}(t) = (1 + \theta t)^{-1/\theta} \quad (3)$$

and the inverse Laplace-Stieltjes transform of the generator φ^{-1} is

$$\mathcal{LS}(\varphi^{-1}) = F(v) = \frac{(1/\theta)^{1/\theta}}{\Gamma(1/\theta)} e^{-v/\theta} \cdot v^{1/\theta-1} \quad (4)$$

The algorithm for sampling from Clayton copula and empirical margins is concluded as in Algorithm 2.

Algorithm 2. $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) = \text{Calyton_sample}(k)$

Step1. Simulate $v \sim F(v) = \frac{(1/\theta)^{1/\theta}}{\Gamma(1/\theta)} e^{-v/\theta} \cdot v^{1/\theta-1}$.

Step2. Simulate i.i.d. $v_i \sim U[0,1]$, $i=1, \dots, n$, get u_i from

$$u_i = \varphi^{-1}((-\log v_i) / v) = (1 - \frac{\theta \log v_i}{v})^{-1/\theta} \quad (5)$$

Step3. Get new individual $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$ by calculating

$$x_i^{(k)} = F_i^{-1}(u_i) = \begin{cases} \text{rand}[x_i^{<j>}, x_i^{<j+1>}] & \text{if } \lfloor u_i \times s \rfloor = j \text{ and } x_i^{<j>} \neq x_i^{<j+1>} \\ \text{rand}(x_i^{<j>}; \delta) & \text{if } \lfloor u_i \times s \rfloor = j \text{ and } x_i^{<j>} = x_i^{<j+1>} \end{cases} \quad (6)$$

3.3 Gumbel Copula Model Sampling

The generator of Gumbel copula is

$$\varphi(t) = (-\ln t)^\theta \quad (7)$$

and the algorithm for sampling from Gumbel copula and empirical margins is described in Algorithm 3.

Algorithm 3. $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) = \text{Gumbel_sample}(k)$

Step1. Simulate an uniform variable $\Theta \sim U(-\frac{\pi}{2}, \frac{\pi}{2})$.

Step2. Simulate an exponentially distributed variable W with mean 1 independently of Θ .

Step3. Set $\alpha = \frac{1}{\theta}$, $\beta = 1$, $\gamma = \left(\cos(\frac{\pi}{2\theta}) \right)^\theta$, $\delta = 0$, and $\theta_0 = \arctan(\beta \tan(\pi\alpha/2)) / \alpha$.

Step4. Compute $Z \sim st(\alpha, \beta, 1, 0)$.

$$\begin{aligned}
 Z &= \frac{\sin \alpha(\theta_0 + \Theta)}{(\cos \alpha\theta_0 \cos \Theta)^{1/\alpha}} \left[\frac{\cos(\alpha\theta_0 + (\alpha-1)\Theta)}{W} \right]^{\left(\frac{1-\alpha}{\alpha}\right)} & \alpha \neq 1 \\
 Z &= \frac{2}{\pi} \left[\left(\frac{2}{\pi} + \beta\Theta \right) \tan \Theta - \beta \ln \left(\frac{\frac{\pi}{2} W \cos \Theta}{\frac{\pi}{2} + \beta\Theta} \right) \right] & \alpha = 1
 \end{aligned} \tag{8}$$

Step5. Compute $v \sim st(\alpha, \beta, \gamma, \delta)$:

$$\begin{aligned}
 v &= \gamma Z + \delta & \alpha \neq 1 \\
 v &= \gamma Z + \left(\delta + \beta \frac{2}{\pi} \gamma \ln(\gamma) \right) & \alpha = 1
 \end{aligned} \tag{9}$$

Step6. Simulate *i.i.d.* $v_i \sim U[0,1]$, $i=1, \dots, n$, get u_i from

$$u_i = \phi^{-1}((- \log v_i) / v) = (1 - \frac{\theta \log v_i}{v})^{-1/\theta} \tag{10}$$

Step7. Get new individual $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$ by calculating

$$\begin{aligned}
 j &= \lfloor u_i \times s \rfloor; \\
 x_i^{(k)} &= F_i^{-1}(u_i) = \begin{cases} \text{rand}[x_i^{<j>}, x_i^{<j+1>}] & , x_i^{<j>} \neq x_i^{<j+1>} \\ \text{rand}(x_i^{<j>}; \delta) & x_i^{<j>} = x_i^{<j+1>} \end{cases}
 \end{aligned} \tag{11}$$

4 Experiments

4.1 Experimental Design

Test Function. To carry out the validation of the proposed algorithm in this paper, we have selected four benchmark functions in Table 1. F_1 and F_2 can easily produce ill-posed covariance matrices; F_3 is a multi-peak function; F_4 is an uni-model function. All of the functions above with the different characteristics are also used in [21].

Initial Population. Because the accuracy of the estimated model and the quality of offspring will decrease as population size decreases, the population size should not too small. For the proposed algorithm, population size is set to 2000, and equally divided into two subpopulations as certain strategy. The dimension of the search space is 10. In order to ensure the fairness of comparison, all the algorithms to participate comparison share common populations. In addition, selection rate of the promising population is 0.5 that truncation selection and roulette selection are 30% and 70% of total selection respectively.

Table 1. Test functions chosen for experiments

Function:	$F_1(x) = -\{10^{-5} + \sum_{i=1}^n y_i \}^{-1}$ where: $y_1 = x_1, y_i = y_{i-1} + x_i, i = 1, \dots, 10$
Search space:	$-0.16 \leq x_i \leq 0.16, i = 1, \dots, 10$
Minimum value:	$F(0) = 10^{-5}$
Function:	$F_2(x) = \sum_{i=1}^n [(x_1 - x_i)^2 + (x_i - 1)^2]$
Search space:	$-5.12 \leq x_i \leq 5.12, i = 1, \dots, 10$
Minimum value:	$F(0) = 0$
Function:	$F_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Search space:	$-5.12 \leq x_i \leq 5.12, i = 1, \dots, 10$
Minimum value:	$F(0) = 0$
Function:	$F_4(x) = \sum_{i=1}^D x_i^2$
Search space:	$-100 \leq x_i \leq 100, i = 1, \dots, 10$
Minimum value:	$F(0) = 0$

Parallel Algorithm Model. The coarse-grained model is used in this paper. Migration topology is unidirectional ring, migration rate and frequency is 0.2 and 5, respectively.

Parameters of Copula Function. The parameter θ of Gumbel is set to 1.05 in Gumbel copula EDA and multi-population parallel copula EDA; the parameter θ of Clayton is set to 1 in Clayton copula EDA and multi-population parallel copula EDA.

4.2 Results Analysis

Using $F_1 - F_4$ test multi-population parallel copula EDA, MIMIC_c^G, UMDA_c^G, Gumbel copula EDA and Clayton copula EDA 50 times respectively, and the evolutionary generation is 150. The experimental results are shown in Table 2. The evaluation criterions are: the mean fitness, the standard variance and the minimal fitness of the 50 optimization results, the average runtime of the 50 optimization.

From the experimental results in table 2, we know that multi-population parallel copula EDA performs well in $F_1 - F_3$ where multivariate correlations are strong. MIMIC_c^G and UMDA_c^G have good performances in F_4 where variables are independent.

To sum up, facing problems of multivariate strong correlations, UMDA_c^G converges very fast and does not converge to the global optimum because the algorithm considers the relationships of the optimized variables as independent, MIMIC_c^G considers the pairwise correlations of the optimized variables, but it consumes much time to get a

little better optimization result. Multi-population parallel copula EDA considers the entire relationships of the optimized variables and converges to an acceptable optimized result in a short time. \square

Table 2. The performances of multi-population parallel copula EDA and some evolutionary algorithms

Algorithm	Mean	Std. Var.	Min	Runtime (m)
F_1				
UMDA _C ^G	-10523.54	11957.07	-60809.21	6.08
MIMIC _C ^G	-44578.33	36613.73	-99780.88	46.58
Gumbel copula EDA	-87975.65	7197.21	-98295.14	27.83
Calyton copula EDA	-81053.08	12160.57	-97704.43	24.32
multi-population	-83290.67	11504.29	-97411.20	22.18
parallel copula EDA				
F_2				
UMDA _C ^G	9.42E-4	3.53E-4	4.27E-4	6.19
MIMIC _C ^G	1.28E-4	6.80E-5	1.97E-5	47.11
Gumbel copula EDA	7.90E-6	4.36E-6	1.24E-6	19.40
Calyton copula EDA	3.37E-6	1.04E-6	1.52E-6	17.11
multi-population	2.78E-6	8.25E-7	1.21E-6	16.70
parallel copula EDA				
F_3				
UMDA _C ^G	8.96	9.77	1.65E-9	4.58
MIMIC _C ^G	9.22	9.39	8.63E-10	46.83
Gumbel copula EDA	6.45E-9	4.24E-9	9.46E-10	14.08
Calyton copula EDA	5.72E-8	2.40E-8	4.96E-9	11.40
multi-population	6.06E-9	5.11E-9	1.01E-9	10.54
parallel copula EDA				
F_4				
UMDA _C ^G	1.66E-9	8.42E-10	3.89E-10	3.73
MIMIC _C ^G	1.39E-9	1.16E-9	4.03E-10	29.90
Gumbel copula EDA	4.26E-9	3.15E-9	1.03E-9	13.30
Calyton copula EDA	6.59E-8	2.97E-8	6.63E-9	11.79
multi-population	5.93E-9	4.13E-9	8.98E-10	10.53
parallel copula EDA				

5 Conclusions

EDAs have effective performance for complex optimization problem by estimating probabilistic distribute model which reflect the dependent relationship of variables in the population. The copula EDAs provides an alternative way to tackle the drawbacks which are modeling complexity, high time consuming and poor performance on problems with strong correlation of variables in traditional EDAs. Multi-population parallel copula EDA joins the idea of parallel GAs with copula EDA. The global population is divided into several independent subpopulations. Each subpopulation takes evolution on different copula under the frame work of copula EDA. And for the purpose of enriching diversity of population and enhancing overall searching ability,

the migration of some individuals among the different subpopulations is allowed under given conditions. The experimental results validate this algorithm.

The initialization and division mechanism for population and the way for multi-population parallel working influence the performance of multi-population parallel copula EDA. The next target is to study how to select a copula function exactly fitting the subpopulation.

Acknowledgments. This work is partially supported by the Youth Science Fund of Taiyuan University of Technology (No.K201031), the Youth Research Fund of ShanXi Province (No. 2010021017-2) and the Science Fund of Shanxi Institution of Higher Learning (No. 2010015).

References

1. Zhou, S.D., Sun, Z.Q.: A Survey on Estimation of Distribution Algorithms. *Acta Automatica Sinica* 33(2), 114–121 (2007)
2. Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Rep. CMU-CS-94-163. Carnegie Mellon University, Pittsburgh, PA (1994)
3. Muhlenbein, 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)
4. De Bonet, J.S., Isbell, C.L., Viola, P.: MIMIC: Finding Optima by Estimation Probability Densities. In: *Advances in Neural Information Processing Systems*, pp. 424–430. MIT Press, Cambridge (1997)
5. Pelican, M., Muhlenbein, H.: The Bivariate Marginal Distribution Algorithm. In: *Advances in Soft Computing-Engineering Design and Manufacturing*, pp. 521–535. Springer, London (1999)
6. Harik, G.: Linkage Learning via Probabilistic Modeling in the ECGA. Illegal Rep. No.99010, Illinois Genetic Algorithms Lab. University of Illinois, Urbana-Champaign, Illinois (1999)
7. Muhlenbein, H., Mahnig, T.: FDA- a Scalable evolutionary Algorithm for the Optimization of Additively Decomposed Functions. *Evolutionary Computation* 7(4), 353–376 (Winter 1999)
8. Pelikan, M., Goldberg, D.E., Cantu-Paz, E.: BOA: the Bayesian Optimization Algorithm. In: *Proc. Genetic and Evolutionary Computation Conference (GECCO-1999)*, Orlando, FL, pp. 525–532 (1999)
9. Larranaga, P., Etxeberria, R., Lozano, J.A., Pena, J.M.: Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2000, Las Vegas, Nevada, USA, July 8-12*, pp. 201–204. Morgan Kaufmann, San Francisco (2000)
10. Sebag, M., Ducoulombier, A.: Extending Population-Based Incremental Learning to Continuous Search Spaces. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) *PPSN 1998. LNCS*, vol. 1498, pp. 418–427. Springer, Heidelberg (1998), <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.1884>

11. Larranaga, P., Lozano, J.A., Bengoetxea, E.: Estimation of Distribution Algorithms based on Multivariate Normal a Gaussian Networks. Technical Report KZZA-IK-1-01. Department of Computer Science and Artificial Intelligence, University of the Basque Country (2001)
12. Wang, L.F., Zeng, J.C., Hong, Y.: Estimation of Distribution Algorithm Based on Archimedean Copulas. In: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GECS 2009), Shanghai, China, June 12-14, pp. 993–996 (2009)
13. 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 (CEC 2009), Trondheim, Norway, May 18-21, pp. 1057–1063 (2009)
14. Nelsen, R.B.: An introduction to copulas, 2nd edn. Springer, New York (2006)
15. Cantú-Paz, E.: A Survey of Parallel Genetic Algorithms. Illinois Genetic Algorithms Laboratory, Urbana-Champaign (1996)
16. Cantú-Paz, E.: Efficient and accurate parallel genetic algorithms. Kluwer, Dordrecht (2001)
17. Wang, L., Guo, X., Zeng, J., Hong, Y.: Using Gumbel Copula and Empirical Marginal Distribution in Estimation of Distribution Algorithm. In: 2010 Third International Workshop on Advanced Computational Intelligence (IWACI 2010), Suzhou, China, August 25-27, pp. 583–587 (2010); (EI:20104613386525)
18. Wang, L., Wang, Y., Zeng, J., Hong, Y.: An estimation of distribution algorithm based on clayton copula and empirical margins. In: Li, K., Li, X., Ma, S., Irwin, G.W. (eds.) LSMS 2010, Part 1. CCIS, vol. 98, pp. 82–88. Springer, Heidelberg (2010), doi:10.1007/978-3-642-15859-9_12, (EI:20104513368882) 04
19. Marshall, A.W., Olkin, I.: Families of Multivariate Distributions. Journal of the American Statistical Association 83, 834–841 (1988)
20. Melchiori, M.R.: Tools For Sampling Multivariate Archimedean Copulas. Yield Curve (April 2006), SSRN: <http://ssrn.com/abstract=1124682>
21. Dong, W., Yao, X.: Unified Eigen Analysis On Multivariate Gaussian Based Estimation of Distribution Algorithms. Information Science 178, 3000–3023 (2008)