Multivariate Gaussian Copula in Estimation of Distribution Algorithm with Model Migration

Martin Hyrš
Brno University of Technology
Faculty of Information Technology
Božetěchova 1/2, Brno, Czech Republic
ihyrs@fit.vutbr.cz

Josef Schwarz
Brno University of Technology
Faculty of Information Technology
Božetěchova 1/2, Brno, Czech Republic
schwarz@fit.vutbr.cz

***Abstract*—The paper presents a new concept of an island-based model of Estimation of Distribution Algorithms (EDAs) with a bidirectional topology in the field of numerical optimization in continuous domain. The traditional migration of individuals is replaced by the probability model migration. Instead of a classical joint probability distribution model, the multivariate Gaussian copula is used which must be specified by correlation coefficients and parameters of a univariate marginal distributions. The idea of the proposed Gaussian Copula EDA algorithm with model migration (GC-mEDA) is to modify the parameters of a resident model respective to each island by the immigrant model of the neighbour island. The performance of the proposed algorithm is tested over a group of five well-known benchmarks.**

I. Introduction

EDAs belong to the advanced evolutionary algorithms based on the estimation and sampling of graphical probabilistic models. They do not suffer from the disruption of building blocks known from the theory of standard genetic algorithms. The canonical sequential EDA is described in Fig. 1.

The important advantage of EDAs is their capability to represent an existing dependency among the variables in individuals using a joint probability distribution.

```
Set $t \leftarrow 0$;
Generate initial population $D(0)$ with $N$ individuals;
while termination criteria is false do
begin
        Select a set $D_s(t)$ of $K < N$ promising individuals;
        Construct the probability model $M$ from $D_s(t)$;
        Sample offspring $O(t)$ from $M$;
        Evaluate $O(t)$;
        Create $D(t+1)$ as a subset of $O(t) \cup D(t)$
          with cardinality $N$;
        $t \leftarrow t + 1$;
end
```

Fig. 1. The pseudocode of canonical EDA.

The EDAs' algorithms can be assorted according to the complexity of the probability model. The well-known EDAs are UMDA algorithms [13], BMDA [12], MIMIC [3], and BOA [11], in the discrete domain. EDAs for the both discrete and continuous domains are described thoroughly in [7]. The main advantage of these algorithms is the capability to discover the variable linkage which results in a successful solution of the complex optimization problem. But there are two problems which must be taken into consideration. The first problem is the need of the model complexity option and its relationship to the probability model overspecification. The second one is the time complexity of the probability model design.

The copula theory has mostly been utilized in the financial and statistical areas [2], [10]. In only recent years the copula theory was imported into the probability model of EDAs [1], [14], [18], [20]. Simply expressed copulas join univariate distribution functions to make the multivariate distribution functions. A copula EDA algorithm therefore has the capability to reduce the execution time and the variable dependency can be modeled more precisely.

The paper is organized as follows. In Section II the basics of the copula theory is presented. The structure of Gaussian copula EDA is described in Section III. In Section IV model migration is described. The experimental results are shown in Section V. In Section VI the obtained results are discussed. The conclusions are provided in Section VII.

II. Copula theory

The copula concept was introduced in 1959 by Sklar [16] in order to separate the effect of dependence from the effect of marginal distributions in a joint distribution. A copula is a function which joins the univariate distribution function and creates multivariate distribution functions. This approach allows us to transform multivariate statistic problems on the univariate problems with the relation represented by just the copula.

A copula C is a multivariate probability distribution function for which the marginal probability distribution of each variable is uniform in $[0, 1]$.

Definition. A copula is a function $C : [0,1]^d \rightarrow [0,1]$ with the following properties [17]:

1) $C(u_1, u_2, \dots, u_d) = 0$ for at least one $u_i = 0$
2) $C(1, 1, \dots, 1, u_i, 1, \dots, 1) = u_i$ for all $i = 1, 2, \dots, d$
3) $\forall (u_1, \dots, u_d), (v_1, \dots, v_d) \in [0,1]^d,\ u_i \leq v_i :$ $\sum_{(w_1,\dots,w_d) \in \times_{i=1}^{d} \{u_i, v_i\}} (-1)^{|\{i | w_i = u_i\}|} C(w_1, \dots, w_d) \geq 0$

This work was supported by the Brno University of Technology project FIT-S-14-2297.

978-1-4799-5375-2/14/$31.00 ©2014 IEEE

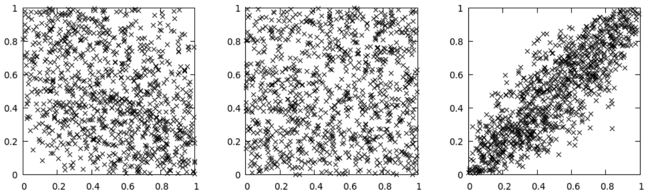

Fig. 2. Scatterplot of samples from bivariate Gaussian copula functions with $\rho = -0.5$ (left, mean negative dependence), $\rho = 0$ (middle, independence), $\rho = 0.9$ (right, strong positive dependence).

Sklar's theorem. Let F be a d-dimensional distribution function with margins $F_1, \ldots, F_d$. Then there exists a d-dimensional copula C such that for all $(x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$F(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_d(x_d)\right) \tag{1}$$

If $F_1, \ldots, F_d$ are continuous, then C is unique. Conversely, if C is a d-dimensional copula and $F_1, \ldots, F_d$ are univariate distribution functions, then the function F defined by (1) is a d-dimensional join distribution function [16].

A. Gaussian copula

The Gaussian copula belonging to the elliptical copula family is associated with the joint normal distribution. The multivariate Gaussian copula has the following functional form:

$$C_R(u_1, \ldots, u_d) = \Phi_R\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\right) \tag{2}$$

where $\Phi_R(x_1, \ldots, x_d) = P(X_1 < x_1, \ldots, X_d < x_d) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_d} \frac{1}{\sqrt{(2\pi)^d |R|}} \,\mathrm{e}^{\left(-\frac{1}{2}(t_1 \ldots t_d) R^{-1} \begin{pmatrix} t_1 \\ \vdots \\ t_d \end{pmatrix}\right)} \,\mathrm{d}t_d \ldots \mathrm{d}t_1$ is the joint multivariate normal cumulative distribution with a zero mean and with a positive-semidefinite correlation matrix R and $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} \,\mathrm{e}^{-\frac{t^2}{2}} \,\mathrm{d}t$ is cumulative distribution function of the standard normal distribution, $\Phi^{-1}(x)$ is its quantile function.

The Gaussian copula can express a correlation in the range $[-1, 1]$ that indicates a perfect negative dependence, through independence to a perfect positive dependence. Examples of bivariate Gaussian copula functions with different correlation matrices $R = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ can be seen in Fig. 2.

III. Gaussian copula EDA

Specifying the Gaussian copula presumes to specify the correlation coefficient and parameters of the univariate marginal distribution. We used a normal distribution as marginal distribution functions.

Similar to the EDA algorithm in Fig. 1, the core of Gaussian EDA includes the selection of promising individuals, identification of the probability model on the level of parameters and a mechanism of the model sampling. The choice of sampling technique depends on the number of copula dimensions so we introduce the methodology for bivariate and multivariate cases.

A. Selection of promising solutions

In order to get promising solutions, we select K best solutions of $D_s(t)$ according to their fitness value. The cardinality K of $D_s(t)$ influences the level of selection pressure and the accuracy of the model.

B. Identification of copula probability model

The probability model is specified by two parts: parameters of marginal distribution functions and parameters of copula. These parameters are derived from the set $D_s(t)$ of selected promising solutions.

For the parameterization of marginal distributions in each dimension i, we used mean value μ_i and standard deviation σ_i. The marginal univariate distribution function is expressed by

$$F_i(x_i) = \int_{-\infty}^{x_i} \frac{1}{\sigma\sqrt{2\pi}} \,\mathrm{e}^{-\frac{(t-\mu_i)^2}{2\sigma_i^2}} \,\mathrm{d}t = \Phi\left(\frac{x_i - \mu_i}{\sigma_i}\right) \tag{3}$$

The Gaussian copula function is parameterized by correlation matrix R. For correlation coefficients in this matrix we used Spearman's correlation coefficient $R_{i,j} = {\rho_S}^{i,j}$.

The Spearman's ${\rho_S}^{i,j}$ between variable instances $x_i = \{x_{i1}, \ldots, x_{in}\}$, $x_j = \{x_{j1}, \ldots, x_{jn}\}$ is computed from the ranked variables. For a sample of size n, the n original values x_{ik}, x_{jk} are converted to ranks r_{ik}, r_{jk}, and ${\rho_S}^{i,j}$ is computed from these:

$${\rho_S}^{i,j} = 1 - \frac{6\sum_{k=1}^{n}(r_{ik} - r_{jk})^2}{n(n^2-1)} \tag{4}$$

C. Sampling offspring from bivariate copula

The main task is to obtain the copula sample $(u, v) \sim C$, then due to the virtue of Sklar's theorem the new values of variables x_1, x_2 can be determined using the inverse of the marginal distribution

$$x_1 = F_1^{-1}(u) \qquad\qquad x_2 = F_2^{-1}(v) \tag{5}$$

In [8] general copula sampling methodology in a low dimensions based on conditional distribution is published. In this methodology, one variable is sampled independently, $u \sim U(0,1)$. It is computed conditional distribution $F_{V|U=u}(v) = \frac{\partial}{\partial u}C(u,v)$ and the inverse of this distribution is $F_{V|U=u}^{-1}(\omega)$. Then it is randomly generating an independent value $\omega \sim U(0,1)$ and variable v is calculated using $v = F_{V|U=u}^{-1}(\omega)$.

Let us start with the partial derivative of the bivariate Gaussian copula distribution

$$\omega = \frac{\partial}{\partial u} C_\rho(u, v) =$$
$$\frac{\partial}{\partial u} \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} \,\mathrm{e}^{-\frac{s^2+t^2-2\rho st}{2(1-\rho^2)}} \,\mathrm{d}t\,\mathrm{d}s \tag{6}$$

After some proper transformations

$$\omega = \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^2}} \,\mathrm{e}^{-\frac{\left(t-\rho\Phi^{-1}(u)\right)^2}{2(1-\rho^2)}} \,\mathrm{d}t \tag{7}$$

Then using inverse operation we get v as

$$v = \Phi\left(\sqrt{1-\rho^2}\Phi^{-1}(\omega) + \rho\Phi^{-1}(u)\right) \tag{8}$$

The couple (u, v) obtained by this computation is the required copula sample. So generating the new offspring from the copula probability model has the following three steps:

1) Randomly generate variables $u \sim U(0,1)$ and $\omega \sim U(0,1)$.
2) Calculate variable $v = \Phi\left(\sqrt{1-\rho^2}\Phi^{-1}(\omega) + \rho\Phi^{-1}(u)\right)$.
3) Determine $x_1 = F_1^{-1}(u)$ and $x_2 = F_2^{-1}(v)$. In our case, as we use univariate marginal normal distribution, the previous equations have form $x_1 = \sigma_1\Phi^{-1}(u) + \mu_1,\ x_2 = \sigma_2\Phi^{-1}(v) + \mu_2$.

D. Sampling offspring from multivariate copula

General bivariate copula sampling methodology used in previous section cannot be used in cases of n-dimensional copulas. Therefore, we used a different approach, based on rejection sampling.

Rejection sampling (also known as acceptance-rejection) is one of the Monte Carlo methods – random numbers are repeatedly generated and checked whether they satisfy the desired distribution. In order to obtain sample x of probability distribution F with density f bounded to interval $[a, b]$ with maximum possible value $m = f(x_m),\ x_m \in [a, b] :\ m \geq f(x),\ \forall x \in [a, b]$ the algorithm shown in Fig. 3 can be used. This approach can be easily extended to multiple dimensions.

But this algorithm uses a density function, while copulas are defined as a cumulative distribution function. Probability density can be derived from cdf by its derivative:

$$\begin{aligned} c_R(u_1,\dots,u_d) &= \frac{\partial^d}{\partial u_1 \dots \partial u_d} C_R(u_1,\dots,u_d) = \\ &= \frac{\partial^d}{\partial u_1 \dots \partial u_d} \int\limits_{-\infty}^{\Phi^{-1}(u_1)} \cdots \int\limits_{-\infty}^{\Phi^{-1}(u_d)} \frac{1}{\sqrt{(2\pi)^d |R|}} \\ &\quad \mathrm{e}^{\left(-\frac{1}{2}(t_1\ \dots\ t_d) R^{-1} \begin{pmatrix} t_1 \\ \vdots \\ t_d \end{pmatrix}\right)} \mathrm{d}t_d \dots \mathrm{d}t_1 = \\ &= \frac{1}{\sqrt{|R|}} \mathrm{e}^{\left(-\frac{1}{2}\left(\Phi^{-1}(u_1)\ \dots\ \Phi^{-1}(u_d)\right)\left(R^{-1}-I\right) \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_d) \end{pmatrix}\right)} \end{aligned} \tag{9}$$

Let us denote matrix $A = \left(R^{-1} - I\right)$ and row/column vector $v = \left(\Phi^{-1}(u_1)\ \dots\ \Phi^{-1}(u_d)\right)$. Matrix A is constant during each evolution generation. Matrix multiplication vAv^T can be effectively computed as

$$\begin{aligned} \left(\Phi^{-1}(u_1)\ \dots\ \Phi^{-1}(u_d)\right)\left(R^{-1} - I\right) \begin{pmatrix} \Phi^{-1}(u_1) \\ \vdots \\ \Phi^{-1}(u_d) \end{pmatrix} &= \\ = vAv^T = \sum_{i=1}^{d}\sum_{j=1}^{d} A_{ij} v_i v_j \end{aligned} \tag{10}$$

```
repeat
    Generate random $x \sim U(a,b)$;
    Generate random $u \sim U(0,1)$;
until $u \leq \frac{f(x)}{m}$;
Return $x$ as desired sample $x \ \sim F$;
```

Fig. 3. The pseudocode of acceptance-rejection sampling.

With this notation, the density of the Gaussian copula is

$$c_R(u_1, ..., u_d) = \frac{1}{\sqrt{|R|}} \mathrm{e}^{\left(-\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} A_{ij} v_i v_j\right)} \tag{11}$$

The maximum value of this density function is $m = \frac{1}{\sqrt{|R|}}\mathrm{e}^0$, i.e. rejection condition in sampling algorithm can be expressed (with the notation used above)

$$u \leq \mathrm{e}^{\left(-\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} A_{ij} v_i v_j\right)} \tag{12}$$

When the copula sample $(u_1, \dots, u_d) \sim C_R$ is given, values of new individual can be obtained by using the inverse of marginal distribution. So generating new offspring from multivariate Gaussian copula probability model has the following steps:

1) Generate random vector $(u_1, \dots, u_d) \sim U(0,1)^d$.
2) Generate random number $u \sim U(0,1)$.
3) Check whether $u \leq \mathrm{e}^{\left(-\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} A_{ij} v_i v_j\right)}$, if not goto step 1.
4) Determine values of new individual by using the inverse of the marginal distribution
$$x_i = F_i^{-1}(u_i) = \sigma_i \Phi^{-1}(u_i) + \mu_i \tag{13}$$

IV. Island-based GC-mEDA

The principal motivation of the new concept of GC-mEDA parallelization developed by our team is to discover the efficiency of the transfer of probabilistic parameters instead of the traditional transfer of individuals. The main goal is to improve algorithm convergence. In the case of EDAs only a few papers deal with the discrete probability model migration [4], [5], [15]. In case of copula-based EDAs, we did not find any papers dealing with model migration which had been published before.

A. EDA with model migration

With the concordance of experimental work done in [15], and according to our experimental results, we used the island-based communication model with bidirectional ring topology. We have investigated several topologies. The bidirectional topology has shown the best performance – it provides a good local interaction and allows propagation of information along the ring in a few steps.

The evolution process on every island runs independently. When the migration condition is met the communication (transfer of model parameters) between the adjacent neighbour islands preceding and succeeding in the bidirectional ring topology is activated. The pseudocode of GC-mEDA algorithm is described in Fig. 4.

```
For each island do in parallel
begin
    Set $t \leftarrow 0$;
    Generate initial population $D(0)$ with $N$ individuals;
    while termination criteria is false do
    begin
        Select a set $D_s(t)$ of $K < N$ promising individuals;
        Construct the copula probability model $M$ from $D_s(t)$;
        if sending condition is met then
        begin
            Send model;
        end
        if immigrant model $M^I$ is received then
        begin
            Combine models: $M^{new} = M^{old} \circ M^I$;
        end
        Sample offspring $O(t)$ from $M$;
        Evaluate $O(t)$;
        Create $D(t+1)$ as a subset of $O(t) \cup D(t)$
          with cardinality $N$;
        $t \leftarrow t + 1$;
    end
end
```

Fig. 4. The pseudocode of GC-mEDA.

B. Model combination

We have decomposed the migration process into pairwise interactions of two islands – one of them is the resident island specified by resident probabilistic model M_R and the other one is the immigrant island whose probabilistic model M_I is transferred to the resident model using a predefined migration rate.

We focused on the problem of combining the immigrant model with the model in the resident island. In general, the modification of the resident model by the immigrant model can be formalized by

$$M_R^{new} = (1-\beta)M_R + \beta M_I \tag{14}$$

where coefficient $\beta \in [0,1]$ specifies the influence of the immigrant model.

We have proposed the following model combination rules:

- Learning of the mean value μ_i of each univariate marginal distribution $F_i(x_i)$
$$\mu_i^{new} = (1-\beta)\mu_i^R + \beta\mu_i^I \tag{15}$$
- Learning of the standard deviation σ_i of each univariate marginal distribution $F_i(x_i)$
$$\sigma_i^{new} = \sqrt{(1-\beta)\left(\left(\mu_i^{new}-\mu_i^R\right)^2 + \left(\sigma_i^R\right)^2\right) + +\beta\left(\left(\mu_i^{new}-\mu_i^I\right)^2 + \left(\sigma_i^I\right)^2\right)} \tag{16}$$
- Learning of the correlation matrix value R_{ij} of Gaussian copula C_R
$$R_{ij}^{new} = (1-\beta)R_{ij}^R + \beta R_{ij}^I \tag{17}$$

We have calculated the coefficient β as

$$\beta = \begin{cases} \frac{fit^R}{fit^R + fit^I} & fit^I \le fit^R \\ 0.1 & \text{otherwise} \end{cases} \tag{18}$$

where fit^R or fit^I represents the average fitness value of set of promising individuals of the resident or the immigrant subpopulation.

V. Experimental works

A. Specification of benchmarks

Five well-known benchmarks from the area of numerical optimization according to [19] are used.

1) Shifted Elliptic Function ($x_i \in [-100, 100]$):
$$f(\mathbf{z}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} z_i^2 \tag{19}$$

2) Shifted Rastrigin's Function ($x_i \in [-5, 5]$):
$$f(\mathbf{z}) = \sum_{i=1}^{D} \left(z_i^2 - 10\cos(2\pi z_i) + 10\right) \tag{20}$$

3) Shifted Ackley's Function ($x_i \in [-32, 32]$):
$$f(\mathbf{z}) = -20\,\mathrm{e}^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}} - \mathrm{e}^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)} + 20 + \mathrm{e} \tag{21}$$

4) Shifted Schwefel's Problem 1.2 ($x_i \in [-100, 100]$):
$$f(\mathbf{z}) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} z_i\right)^2 \tag{22}$$

5) Shifted Rosenbrock's Function ($x_i \in [-100, 100]$):
$$f(\mathbf{z}) = \sum_{i=1}^{D-1} \left(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2\right) \tag{23}$$

D denotes the number of dimensions and $\mathbf{z}$ is the shifted candidate solution $\mathbf{z} = \mathbf{x} - \mathbf{o}_{shift}$. The fitness function equals directly to benchmark function and the minimization task is solved. We used the following settings:

- Population size on every island: 500.
- Number of islands: 10.
- Migration rate: after every 20 generations (i.e. 10,000 fitness evaluations per island).
- Maximum number of fitness evaluations: 3,000,000.
- Selection: We used $K = 0.2N$, i.e. 100 individuals.
- Problem size: 10 variables for all problems.

B. Experiments

Our first experiment was to run sequential version of GC-EDA algorithm which enables to make a comparison with the island based version. In the second experiment the island based version of GC-mEDA was to run up to 3,000,000 fitness evaluation. 20 independent runs were carried out for each minimization task.

The results (best fitness value) from the both experiment for a few evolution epochs are presented in Tables I–V. The basic statistics are presented including mean value, standard deviation and best and worst value of fitness function.

TABLE I. EXPERIMENT RESULTS FOR SHIFTED ELLIPTIC FUNCTION.

fitness eval.	mean	std. dev.	min	max
island based GC-mEDA				
30,000	1.7523e-01	8.2618e-02	1.2459e-08	2.6327e-01
50,000	9.2749e-02	8.3155e-02	9.5495e-09	2.3337e-01
100,000	2.4324e-03	1.0077e-02	3.6928e-10	4.6344e-02
300,000	5.4174e-14	4.3172e-14	9.8561e-15	1.9772e-13
1,000,000	7.4528e-15	5.1897e-15	1.2294e-15	1.8366e-14
3,000,000	6.7061e-15	4.8215e-15	1.2294e-15	1.6128e-14
sequential GC-EDA				
30,000	4.9601e-13	6.2912e-13	3.1955e-15	2.1381e-12
50,000	3.7627e-13	5.6109e-13	2.6323e-15	2.1379e-12
100,000	3.7629e-13	5.6113e-13	2.6323e-15	2.1379e-12
300,000	3.7629e-13	5.6113e-13	2.6323e-15	2.1379e-12
1,000,000	3.7629e-13	5.6113e-13	2.6323e-15	2.1379e-12
3,000,000	3.7629e-13	5.6113e-13	2.6323e-15	2.1379e-12

TABLE II. EXPERIMENT RESULTS FOR SHIFTED RASTRIGIN'S FUNCTION.

fitness eval.	mean	std. dev.	min	max
island based GC-mEDA				
30,000	1.1437e+01	6.3805e+00	1.2992e+00	2.4656e+01
50,000	8.5174e+00	7.0599e+00	3.6694e-01	2.4656e+01
100,000	3.9228e-01	9.0083e-01	1.4879e-06	3.2059e+00
300,000	6.4266e-15	8.8063e-15	8.6736e-19	3.2517e-14
1,000,000	9.3072e-18	2.4880e-17	8.6736e-19	1.1395e-16
3,000,000	9.3072e-18	2.4880e-17	8.6736e-19	1.1395e-16
sequential GC-EDA				
30,000	4.4640e-13	1.5348e-12	1.2271e-14	7.1213e-12
50,000	7.8149e-16	1.2276e-15	1.7347e-18	4.4608e-15
100,000	7.8145e-16	1.2274e-15	1.7347e-18	4.4600e-15
300,000	7.8145e-16	1.2274e-15	1.7347e-18	4.4600e-15
1,000,000	7.8145e-16	1.2274e-15	1.7347e-18	4.4600e-15
3,000,000	7.8505e-16	1.2251e-15	1.7347e-18	4.4600e-15

TABLE III. EXPERIMENT RESULTS FOR SHIFTED ACKLEY'S FUNCTION.

fitness eval.	mean	std. dev.	min	max
island based GC-mEDA				
30,000	8.8701e-03	3.1785e-03	4.9903e-04	1.2050e-02
50,000	5.5196e-03	3.3885e-03	4.9903e-04	1.0963e-02
100,000	1.5471e-03	1.8576e-03	1.9741e-06	5.9593e-03
300,000	4.0409e-09	2.5058e-09	1.3333e-09	1.1597e-08
1,000,000	1.2756e-09	4.4931e-10	4.1203e-10	2.1368e-09
3,000,000	1.0019e-09	2.4738e-10	4.1203e-10	1.3875e-09
sequential GC-EDA				
30,000	5.0010e-09	2.2127e-09	1.3381e-09	1.0255e-08
50,000	3.4812e-09	1.9029e-09	1.1881e-09	1.0204e-08
100,000	3.4772e-09	1.9002e-09	1.1881e-09	1.0204e-08
300,000	3.4772e-09	1.9002e-09	1.1881e-09	1.0204e-08
1,000,000	3.4772e-09	1.9002e-09	1.1881e-09	1.0204e-08
3,000,000	3.4772e-09	1.9002e-09	1.1881e-09	1.0204e-08

VI. RESULTS DISCUSSION

From the comparison in Tables I–V it is evident that the parallel version of the algorithm is better for all benchmarks. Sequential version is able to find relatively good local solution quite fast but then it loses its diversity and no more improvement is done. The parallel version of algorithm converges slowly but it has the capability to find near optimal solution. We suppose that this performance is caused by the phenomenon of the proposed model migration.

TABLE IV. EXPERIMENT RESULTS FOR SHIFTED SCHWEFEL'S PROBLEM 1.2.

fitness eval.	mean	std. dev.	min	max
island based GC-mEDA				
30,000	1.5130e-03	8.7889e-04	3.8398e-06	2.7258e-03
50,000	8.7465e-04	7.3725e-04	9.6912e-08	2.0026e-03
100,000	7.3541e-05	3.1364e-04	2.1006e-11	1.4406e-03
300,000	1.5106e-16	1.3937e-16	3.0368e-17	7.0236e-16
1,000,000	1.1605e-16	5.7798e-17	3.0368e-17	2.1242e-16
3,000,000	1.1605e-16	5.7798e-17	3.0368e-17	2.1242e-16
sequential GC-EDA				
30,000	7.1605e-16	9.9718e-16	4.5433e-17	3.9615e-15
50,000	5.1579e-16	9.3238e-16	4.5433e-17	3.9598e-15
100,000	5.1694e-16	9.3237e-16	4.5433e-17	3.9600e-15
300,000	5.1610e-16	9.3237e-16	4.5433e-17	3.9600e-15
1,000,000	5.1610e-16	9.3237e-16	4.5433e-17	3.9600e-15
3,000,000	5.9737e-16	8.9020e-16	4.5433e-17	3.9600e-15

TABLE V. EXPERIMENT RESULTS FOR SHIFTED ROSENBROCK'S FUNCTION.

fitness eval.	mean	std. dev.	min	max
island based GC-mEDA				
30,000	8.6062e+00	3.2655e-01	7.7566e+00	9.2612e+00
50,000	8.3131e+00	2.5613e-01	7.7566e+00	8.7071e+00
100,000	7.8518e+00	2.1773e-01	7.4629e+00	8.3980e+00
300,000	7.5475e+00	1.2830e-01	7.2701e+00	7.7540e+00
1,000,000	7.4713e+00	1.2751e-01	7.1934e+00	7.7110e+00
3,000,000	7.3781e+00	1.1675e-01	7.1350e+00	7.6116e+00
sequential GC-EDA				
30,000	7.9487e+00	1.9491e-01	7.5773e+00	8.3465e+00
50,000	7.9480e+00	1.9466e-01	7.5767e+00	8.3446e+00
100,000	7.9480e+00	1.9464e-01	7.5767e+00	8.3445e+00
300,000	7.9480e+00	1.9464e-01	7.5767e+00	8.3445e+00
1,000,000	7.9480e+00	1.9464e-01	7.5767e+00	8.3445e+00
3,000,000	7.9480e+00	1.9464e-01	7.5767e+00	8.3445e+00

In Tables VI and VII we arranged a comparison of parallel version of our algorithm with the sequential versions of other published algorithms that used different versions of copula. The comparison is done for the same number of executed fitness evaluations.

In Table VI the comparison with the algorithm using Copula Bayesian network [9] after 100,000 evaluations is shown. For the case of Rastrigin's, Ackly's and Schwefel's 1.2 function GC-mEDA is evidently better, for Elliptical and Rosenbrock's function the results are comparable.

In Table VII the comparison with the other suite of algorithms [6], [21] is carried out on the level of 300,000 evaluations. In case of Rosenbrock's function the results are comparable, for Rastrigin's and Sphere function our algorithm is better (Sphere function is simplified version of Elliptic problem).

It is important to note that we primary used weak model of parallelization, so with the increasing number of island the size of subpopulation is constant and the same as in the sequential version. It follows from the need to keep subpopulation large enough for adequate probability model design. We count the number of evaluations over all islands to keep the computational costs expressed by total number of evaluations comparable with sequential variant. It means that we compared the statistics of the island-based and the

TABLE VI. COMPARISON WITH PAPER [9]. "CBN" DENOTES COPULA BAYESIAN NETWORK FROM [9].

	Ellip.	Rastr.	Ack.	Schw. 1.2	Rosen.
CBN [9]	6.06e-03	2.39e+00	3.71e-02	2.23e+01	1.05e+01
GC-mEDA	2.43e-03	3.92e-01	1.55e-03	7.35e-05	7.85e+00

TABLE VII. COMPARISON WITH PAPERS [6], [21]. "CE" MEANS COPULA EDA, "CE-KS" COPULA EDA OF DYNAMIC K-S TEST FROM [21]; "CL" DENOTES CLAYTON, "GU" GUMBEL, "SN" SN-EDA FROM [6].

	Ellip./sphere	Rastrigin's	Rosenbrock's
cE [21]	4.62e-08	6.45e-08	6.52e+00
cE-KS [21]	1.16e-08	2.60e-08	7.05e+00
Cl [6]	1.45e-07	7.00e-08	8.36e+00
Gu [6]	3.59e-09	5.49e-09	6.62e+00
Sn [6]	1.22e-09	9.52e-09	6.54e+00
GC-mEDA	5.42e-14	6.42e-15	7.55e+00

sequential versions for the same number of evaluations. On the other hand, the parallel version can execute island-count-times higher number of evaluations than the sequential one in the approximately same execution time.

VII. CONCLUSION

In this paper we have introduced the utilization of multivariate Gaussian copula as a case of probability model in Estimation of Distribution Algorithm with model migration. We have presented the main theoretical basis and an approach of constructing and sampling the Gaussian copula. For bivariate copula we have presented the sampling method based on the conditional distribution function, and acceptance-rejection sampling in the case of the multivariate Gaussian copula.

The process of the model migration includes migration of the marginal distribution parameters and the copula function parameters. It allows to combine the resident island probability model with the immigrant probability model to obtain new resident island model.

In order to illustrate the performance of GC-mEDA algorithm, a few known benchmarks of optimization in continuous domain were used. From the experimental results it is clear that the proposed algorithm is effective and competitive with the results of other algorithms published up to now.

Our future research will be focused on the utilization of different types of copulas and the implementation of different univariate marginal distributions and their modification during the evolution process. An additional problem is tuning of the whole learning process and the learning of copula parameters.

REFERENCES

[1] K. Aas, C. Czado, A. Frigessi, and H. Bakken, "Pair-copula constructions of multiple dependence," *Insurance: Mathematics and Economics*, vol. 44, no. 2, pp. 182 – 198, 2009.

[2] U. Cherubini, E. Luciano, and W. Vecchiato, *Copula Methods in Finance*. Hoboken, NJ: John Wiley & Sons, 2004.

[3] J. S. De Bonet, C. L. Isbell, and P. A. Viola, "MIMIC: Finding optima by estimating probability densities," in *Advances in Neural Information Processing Systems*, vol. 9. The MIT Press, Cambridge, 1997, pp. 424–430.

[4] L. delaOssa, J. A. Gámez, and J. M. Puerta, "Migration of probability models instead of individuals: An alternative when applying the island model to edas," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science. Springer, 2004, vol. 3242 of LNCS, pp. 242–252.

[5] ——, "Improving model combination through local search in parallel univariate edas." in *Congress on Evolutionary Computation*, vol. 2. IEEE, 2005, pp. 1426–1433.

[6] B. Jia, L. Wang, and Z. Cui, "Copula for estimation of distribution algorithm based on goodness-of-fit test," in *Journal of Theoretical and Applied Information Technology*, no. 3, 2013, pp. 1128–1132.

[7] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[8] J. Mai and M. Scherer, *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*, ser. Series in quantitative finance. Imperial College Press, 2012, vol. 4.

[9] M. Méndez and R. Landa, "An EDA based on bayesian networks constructed with archimedean copulas," in *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 2012, pp. 188–193.

[10] R. B. Nelsen, *An Introduction to Copulas*, ser. Springer Series in Statistics. Springer New York, 2006.

[11] M. Pelikan, D. Goldberg, and E. Cantú-Paz, "BOA: The bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, vol. I, 1999, pp. 525–532 also IlliGAL Report no. 99003.

[12] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing*. Springer London, 1999, pp. 521–535.

[13] ——, "Marginal distributions in evolutionary algorithms," in *In Proceedings of the International Conference on Genetic Algorithms Mendel 98*, 1999, pp. 90–95.

[14] R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E. R. Villa-Diharce, "Using copulas in estimation of distribution algorithms," in *MICAI 2009: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5845, pp. 658–668.

[15] J. Schwarz and J. Jaroš, "Parallel bivariate marginal distribution algorithm with probability model migration," in *Linkage in Evolutionary Computation*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2008, vol. 157, pp. 3–23.

[16] A. Sklar, "Fonctions de répartition à n dimensions et leurs marges," *Publications de l'Institut de Statistique de l'Université de Paris*, vol. 8, pp. 229–231, 1959.

[17] ——, "Random variables, joint distribution functions, and copulas," *Kybernetika*, vol. 9, no. 6, pp. 449–460, 1973.

[18] M. Soto, Y. González-Fernández, and A. Ochoa, "Modeling with copulas and vines in estimation of distribution algorithms," *CoRR*, vol. abs/1210.5500, 2012.

[19] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization," Nature Inspired Computation and Applications Laboratory, USTC, China, Tech. Rep., 2009. [Online]. Available: http://nical.ustc.edu.cn/cec10ss.php

[20] L.-F. Wang, J.-C. Zeng, and Y. Hong, "Estimation of distribution algorithm based on copula theory," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 1057–1063.

[21] H. Zhao and L. Wang, "Marginal distribution in copula estimation of distribution algorithm based dynamic K-S test," in *IJCSI International Journal of Computer Science Issues*, no. 3, 2012, pp. 507–514.