

Estimation of Distribution Algorithm Based on Copula Theory

Li-Fang Wang and Jian-Chao Zeng

Abstract. Estimation of Distribution Algorithms (EDAs) is the hot topic of evolutionary computation currently. EDAs model the selected population using a distribution model, which is latter sampled to generate the population for the next generation. This chapter introduces a new way to estimate the distribution model and sample from it according to copula theory. The multivariate joint is decomposed into the univariate margins and a function called copula. In the EDAs based on copula theory (copula-EDAs), only the margins are estimated, and the next generation is sampled from the copula and the inverse function of the margins. The framework of the copula-EDAs is discussed in the chapter. Two 2-dimensional copula-EDAs and a high-dimensional copula-EDA are described in detail as the examples.

1 Introduction

Estimation of Distribution Algorithms (EDAs) are originated from Genetic Algorithms (GAs) [1]. While GAs generate a new population using crossover and mutation operators, EDAs estimate a distribution model of the selected population and sample from the estimated model. The comparison of the two algorithms is shown as Fig. 1.

As an example, the integer minimization problem $f(x)=x^2$, $x \in [0,15]$ is optimized with a kind of EDA. The value of x is denoted as 4-bit binary number, *i.e.* 5 is denoted as 0101, 8 is denoted as 1000. According to the flow chat of EDAs, the first step is to generate some random integers in $[0,15]$. Supposing, the following 5 binary numbers are generated. Their fitness are listed followed the numbers.

No.	x	fitness
1	1 0 1 0	100
2	1 1 0 0	144
3	0 1 0 0	16
4	0 0 1 1	9
5	0 1 0 1	25

Li-Fang Wang · Jian-Chao Zeng
Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology, China P.R.
e-mail: wlf1001@163.com

Secondly, a promising population is selected according to the individual’s fitness. If the truncation selection is used and the selection rate is 0.6, then the 3rd-5th individuals are selected as the promising population.

Thirdly, the distribution model of the selected population is estimated. Denote p_i as the probability of the i^{th} bit being 1. Thus the probability of each bit is the following: $p_1=0, p_2=2/3, p_3=1/3, p_4=2/3$.

Fourthly, the next generation is sampled according to the estimated model. The following individuals are generated supposing.

No.	x	fitness
1	0 0 1 0	4
2	0 1 0 1	25
3	0 1 0 0	16
4	0 0 0 1	1
5	0 1 0 1	25

The loop is continued by going to the second step until the termination condition is met.

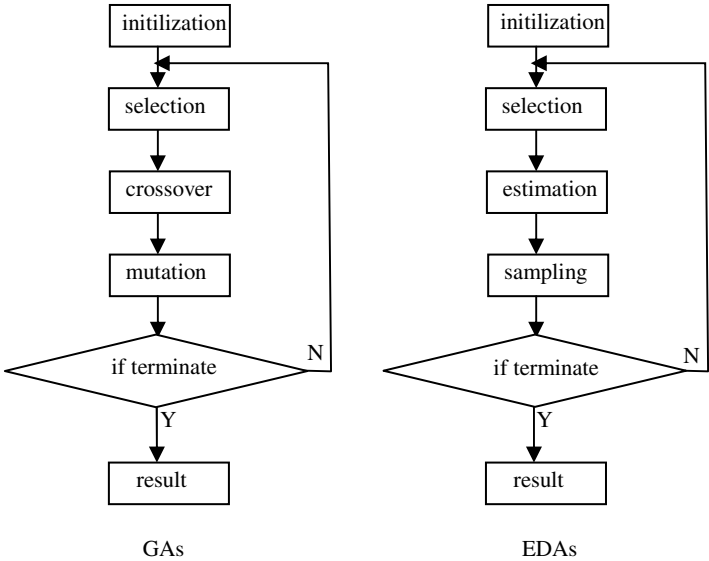


Fig. 1 The different flowchart of EDAs and GAs

Different kinds of EDAs have been developed since the idea of EDA is introduced in 1994 [1,2].

The early EDAs regard the relationship of the random variables as independent. PBIL(Population Based Incremental Learning) [2-5], UMDA(Univariate Marginal Distribution Algorithm) [6-8] and cGA(compact Genetic Algorithm) [9] are all the classical algorithms.

However, in practical problems variables of the optimization problems are not independent. Some algorithms consider the linear relationship of variables. The random variables are numbered according to a certain rule, and the random variable depends only on the previous random variable. The classical algorithms of this type include MIMIC(Mutual Information Maximization of Input Clustering) [10], COMIT(Combining Optimizers with Mutual Information Trees) [11] and BMDA(Binary Marginal Distribution Algorithm) [12,13].

Many EDAs considering the multivariate dependence are proposed in the last decade. For example, ECGA(Extended Compact Genetic Algorithm) [14], FDA(Factorized Distribution Algorithms) [15-19] and BOA(Bayesian Optimization Algorithm) [20-25].

Continuous EDAs were developed later than discrete EDAs. The classical algorithms include PBILc [26], UMDAc [27], SHCLVND(Stochastic Hill Climbing with Learning by Vectors of Normal Distributions) [28], EMNA(Estimation of Multivariate Normal Algorithm)[29], ENGGA(Estimation of Gaussian Networks Algorithm) [30], IDEA [31], etc. The Gaussian distribution is used in most of these algorithms.

PBILc and UMDAc are extended from PBIL and UMDA respectively to the continuous form, and the Gaussian distributions are used in both of the two algorithms. The Gaussian distribution is also used in SHCLVND and the parameter mean is adjusted by Hebbian Learning.

The multivariate Gaussian distribution is used to estimate the distribution of the population in EMNA. The parameters mean and the covariance matrix are estimated by EML. The next generation is sampled from the joint distribution. It is a problem in EMNA that the runtime to estimate the covariance matrix increased exponentially if the number of random variables has increased.

EGNA is an algorithm based on the Gaussian network. The directed edges represent the relationship of the random variables and the distribution of each random variable is described with the Gaussian distribution. The Gaussian network is adjusted according to the current selected population.

IDEA is a kind of EDA based on the hybrid Gaussian distribution and the Gaussian kernel function. The shortage of EMNA and EGNA is conquered in IDEA. But the relationship of random variables is not reflected thoroughly in IDEA.

The chapter describes how copula theory can be used in an Estimation of Distribution Algorithm. The benefit of the copula approach is that only marginal distributions need to be estimated from the population of solutions, thus promising efficiency savings.

The remaining sections are organized as follows. Section 2 talks about the basic theorem and properties in copula theory. Section 3 explains why copula theory could be used in EDA and then gives the framework of copula-EDAs. Section 4 and section 5 illustrate the copula EDAs with two 2-dimensional algorithms and one high-dimensional algorithm respectively. Finally, section 6 concludes the paper, summarizing the main results and pointing further directions.

2 A Brief Introduction to Copula Theory

Copula theory [32] is one of the hot topics in statistics. Its main idea is how to decompose the multivariate joint distribution into the univariate marginal distributions. A kind of function called copula ties the joint and the margins together. Thus the difficult work to estimate the joint distribution could be replaced by estimating the margins and constructing a copula. It also draws many scholars' attention to sample from copula. These are the two key steps in EDAs, i.e., to construct a distribution model to represent the selected population and to sample the constructed model, creating the next generation. The distribution of the population is reflected precisely and the runtime is shortened if the copula theory is introduced into the framework of EDAs. Copula theory is briefly introduced in the following section.

2.1 Definitions and Basic Properties

The definition of copula and the basic theorem are listed in this section. The symbol \mathbf{I} denotes the domain $[0, 1]$ in this chapter. The definitions and theorems are quoted from [32].

Definition 1. A two-dimensional subcopula (or 2-subcopula, or briefly, a subcopula) is a function C' with the following properties:

- 1) $\text{Dom}C' = S_1 \times S_2$, where S_1 and S_2 are subsets of \mathbf{I} containing 0 and 1;
- 2) C' is grounded and 2-increasing;
- 3) For every u in S_1 and every v in S_2 ,

$$C'(u, 1) = u \text{ and } C'(1, v) = v. \quad (1)$$

Definition 2. A two-dimensional copula (or 2-copula, or briefly, a copula) is a 2-subcopula C whose domain is \mathbf{I}^2 .

Equivalently, a copula is function C from \mathbf{I}^2 to \mathbf{I} with the following properties:

- 1) For every u, v in \mathbf{I} ,

$$C(u, 0) = 0 = C(0, v) \quad (2)$$

and

$$C(u, 1) = u \text{ and } C(1, v) = v; \quad (3)$$

- 2) For every u_1, u_2, v_1, v_2 in \mathbf{I} such that $u_1 \leq u_2$ and $v_1 \leq v_2$,

$$C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0. \quad (4)$$

The definition of n -copula is extended from the above definitions.

Definition 3. An n -copula is a function C from \mathbf{I}^n to \mathbf{I} with the following properties:

- 1) For every \mathbf{u} in \mathbf{I}^n ,

$$C(\mathbf{u}) = 0 \text{ if at least one coordinate of } \mathbf{u} \text{ is } 0, \quad (5)$$

and

$$\text{if all coordinates of } \mathbf{u} \text{ are } 1 \text{ except } u_k, \text{ then } C(\mathbf{u}) = u_k; \quad (6)$$

2) For every \mathbf{u} and \mathbf{v} in \mathbf{I}^n such that $\mathbf{u} \leq \mathbf{v}$, denote

$$V_C([\mathbf{u}, \mathbf{v}]) = \sum_{\varepsilon_1, \dots, \varepsilon_n \in \{0,1\}} (-1)^{(\varepsilon_1 + \dots + \varepsilon_n)} C((\varepsilon_1 u_1 + (1 - \varepsilon_1) v_1) + \dots + (\varepsilon_n u_n + (1 - \varepsilon_n) v_n))$$

then

$$V_C([\mathbf{u}, \mathbf{v}]) \geq 0. \quad (7)$$

It can be derived from the definition of copula that the condition of copula is very easy to be satisfied. So many functions can be used as copulas. The copulas are divided into two classes that are elliptical copulas and Archimedean copulas. For example, normal copula and t-copula are all elliptical copulas. Archimedean copulas are generated from different generators such as Clayton copula, Gumbel copula and others. Each Archimedean copula has one or more parameters. More details see [32].

Sklar's theorem plays an important role in copula theory. It gives the theory basis to connect the multivariate distribution with one-dimensional marginal distribution.

Sklar's Theorem in n -dimensions: Let H be an n -dimensional distribution function with margins F_1, F_2, \dots, F_n . Then there exists an n -copula C such that for all \mathbf{x} in \mathbf{R}^n ,

$$H(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n)). \quad (8)$$

If F_1, F_2, \dots, F_n are all continuous, then C is unique; otherwise, C is uniquely determined on $\text{Ran} F_1 \times \dots \times \text{Ran} F_n$. Conversely, if C is an n -copula and F_1, F_2, \dots, F_n are distribution functions, then the function H defined by (8) is an n -dimensional distribution function with margins F_1, F_2, \dots, F_n .

According to Sklar's theorem, the joint distribution of a random vector could be constructed by the one-dimensional distribution of each random variable through a copula. Therefore, the hard work to estimate the joint distribution is simplified to estimate the margins and to produce a copula.

2.2 Random Variable Generation

As it will be demonstrated in section 4 and section 5, it is easy in EDAs to estimate the distribution model based on Sklar's theorem. And the next work is to sample from the distribution model that is composed by a copula and some margins. An example of sampling from 2- copula is shown. Let $F(x)$ and $G(y)$ be the margins of the random vector (X, Y) , and the joint is $C(F(x), G(y))$, i.e. C is the copula. Denote $U = F(x)$, $V = G(y)$, then $U \in [0, 1]$, $V \in [0, 1]$. According to Sklar's theorem, the sample of (X, Y) can be generated by the following two steps. First, a sample value (u, v) of the random vector $(U, V) \in [0, 1]^2$ that obeys the joint C is generated. Second, the sample value (x, y) is generated according to the inverse of margins and (u, v) , i.e. $x = F^{-1}(u)$ and $y = G^{-1}(v)$.

Obviously, the first step need to be studied and it is actually one of the current hot topics for research. A method based on the conditional distribution of $C(u, v)$ is introduced as an example. The condition distribution $c_u(v)$ when

$U=u$ is $c_u(v)=P[V \leq v|U=u]=\lim_{\Delta u \rightarrow 0} \frac{C(u+\Delta u, v)-C(u, v)}{\Delta u} = \frac{\partial C(u, v)}{\partial u}$. So the steps are the following.

- 1) Generate two independent random numbers $u \sim U(0, 1)$ and $t \sim U(0, 1)$;
- 2) Let $v=c_u^{(-1)}(t)$, where, $c_u^{(-1)}(t)$ is the restricted inverse function of $c_u(v)$ in $[0, 1]$;
- 3) (u, v) is the sample that obeys C .

This is a way to sample from the 2-copula based on the conditional distribution and it can be extended to n -copula. Furthermore, many ways to sample from the n -copula are studied by the scholars [33,34].

3 Motivation

The main idea of EDAs is to analyze the distribution model of the promising population and to affect the next generation by use of the distribution model. Selection, modeling and sampling are the three main steps of EDAs. Selection is used to increase the average fitness of the population by eliminating low fitness individuals while increasing the number of copies of high fitness ones. Modeling is used to quantify the distribution of the selected population, *i.e.* by probability density function (pdf), cumulative distribution function (cdf) or some other ways. Sampling is used to generate the next generation according to the modeled distribution.

To reflect the entire relation of the random variables and the distribution model of selected population, the joint of all random variables is the best way. However, the computation cost to model the joint is very large when the problem involves a large number of variables. For instance, EMNA supposes that the joint is multivariate normal distribution, and the parameters (*i.e.* mathematical expectation and covariance matrix) are estimated by EML. Obviously, the size of covariance matrix is the square of the number of variables in the problem. The larger the problem we are trying to solve, the higher is the computational cost to estimate the necessary parameters.

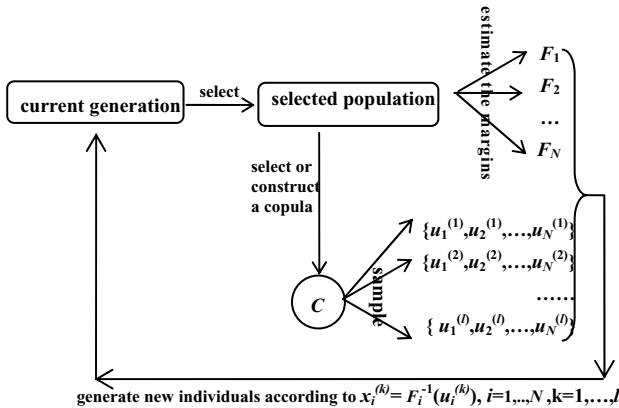


Fig. 2 The schema of copula-EDA

On the grounds of Sklar's theorem, the joint of random variables could be represented by a copula and the margins of each random variable. Thus the relation of random variables could be denoted by a selected copula and the margins are also estimated as a certain distribution, and then the joint is obtained according to equation (8). Actually, the joint H does not need to be calculated in copula-EDAs. The only two works are to select or to construct a copula C and to estimate each one-dimensional margins $F_i, i=1, \dots, N$.

Sampling from the estimated distribution model is the next work after the distribution model of selected population is estimated, that is the new individuals generated should obey the estimated distribution. To generate an individual, the first work is to generate a vector $\{u_1, u_2, \dots, u_N\} \in [0, 1]^N$ who obeys the joint C according to copula theory. Subsequently, the values $x_i = F_i^{-1}(u_i)$ are calculated according to the inverse function of each margins, and the vector $\{x_1, x_2, \dots, x_N\}$ is the sample who obeys the joint H , i.e. $\{x_1, x_2, \dots, x_N\}$ is one generated individual. To sum up the above arguments, the schema of EDAs based on copula theory (copula-EDA) is shown in Fig. 2. Copula-EDA generates random initial population firstly, then repeat the following 4 steps until certain terminate condition is met.

1. Select m individuals from the current population as the promising population.
2. Estimate the margins. The m individuals are the m samples of the N -dimensional random vector. Each one-dimensional margin $F_i (i=1, \dots, N)$ is estimated according to the m samples. The margins could be empirical distribution, normal distribution or others.
3. Sample from the copula C . Generate l vectors $\{u_1^{(j)}, u_2^{(j)}, \dots, u_N^{(j)}\} (j=1, \dots, l)$ who obey the joint C .
4. Compose the next generation by the following 3 parts. 1) Reserve the best k individuals of the current generation to the next generation. 2) Get the new l individuals by calculate $x_i^{(j)} = F_i^{-1}(u_i^{(j)}) (i=1, \dots, N, j=1, \dots, l)$. 3) to generate some random individuals in the search space depending on certain mutation rate.

4 Two-Dimensional Copula-EDAs

Suppose that the optimization problem is

$$\min f(X) = f(x_1, x_2, \dots, x_n), x_i \in [a_i, b_i] \quad (i=1, 2, \dots, n)$$

Let s denote the population size. Let $\mathbf{x} = \{x^i = (x_1^i, x_2^i, \dots, x_n^i), i=1, 2, \dots, s\}$ denote the population. Then the population \mathbf{x} is composed of s observed samples from a random vector (X_1, X_2, \dots, X_n) . The marginal distribution of each random variable $X_i (i=1, 2, \dots, n)$ can be Gaussian distribution, t-distribution or some other distribution. On the ground of Sklar's theorem, the joint distribution function can be constructed with a selected copula and the marginal distributions. The modeling for the distribution of the selected population is finished. Next step is to generate new population by the way introduced in Sect. 2.2.

4.1 Gaussian Copula-EDAs

A two-dimensional optimization problem is considered. According to copula theory, the marginal distribution of each random variable and a copula are required. The marginal distributions of (X_1, X_2) is easy to estimate because the observed values are given. The only thing is to evaluate sample mean and sample variance. 2-D Gauss-copula is selected which is

$$C(u, v; \rho) = \phi_\rho(\phi^{-1}(u), \phi^{-1}(v)), u, v \in [0, 1], \quad (9)$$

where, ϕ_ρ is a binary standard Gaussian distribution with correlation coefficient ρ . Determining ρ is the key to construct copula.

If the Maximum Likelihood Estimation is used, then the log-likelihood function is

$$l(\rho) = \sum_{i=1}^s \left\{ -\ln 2\pi - \frac{1}{2} \ln(1-\rho^2) - \frac{z_{1i}^2 + z_{2i}^2 - 2\rho z_{1i} z_{2i}}{2(1-\rho^2)} \right\} + \sum_{i=1}^s [\ln f(x_1^i) + \ln g(x_2^i)] \quad (10)$$

where,

$$z_{1i} = \phi^{-1}(F(x_1^i)), z_{2i} = \phi^{-1}(G(x_2^i)). \quad (11)$$

If Moment Estimation is used, then

$$\rho = \frac{\frac{1}{s} \sum_{i=1}^s x_1^i x_2^i - \bar{X}_1 \bar{X}_2}{S_{X_1}^* S_{X_2}^*}, \quad (12)$$

where,

$$\bar{X}_k = \frac{1}{s} \sum_{i=1}^s x_k^i, S_{X_k}^* = \sqrt{\frac{1}{s-1} \sum_{i=1}^s (x_k^i)^2}, (k=1, 2). \quad (13)$$

Next, sampling from the distribution function is discussed. Since

$$\begin{aligned} C(u, v; \rho) &= \phi_\rho(\phi^{-1}(u), \phi^{-1}(v)) \\ &= \int_{-\infty}^u \int_{-\infty}^v \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{[\phi^{-1}(s)]^2 + [\phi^{-1}(t)]^2 - 2\rho\phi^{-1}(s)\phi^{-1}(t)}{2(1-\rho^2)}\right\} dt ds \end{aligned} \quad (14)$$

then

$$\begin{aligned} \omega &= C_u(v) = \partial C(u, v) / \partial u \\ &= e^{\frac{[\phi^{-1}(u)]^2}{2}} \int_{-\infty}^v \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{[\phi^{-1}(t) - \rho\phi^{-1}(u)]^2}{2(1-\rho^2)}\right\} dt \end{aligned} \quad (15)$$

Therefore,

$$v = \sqrt{1-\rho^2} \phi^{-1}\left(e^{\frac{[\phi^{-1}(u)]^2}{2}} \omega\right) + \rho\phi^{-1}(u) = \sqrt{1-\rho^2} \phi^{-1}(\omega') + \rho\phi^{-1}(u) \quad (16)$$

So randomly generate vector $(u, \omega) \sim U[0,1]^2$, and then the other random number v is calculated by using equation (16). The vector (x_1, x_2) can be calculated by using equation (17).

$$x_1 = \sigma_1 \phi^{-1}(u) + \mu_1, x_2 = \sigma_2 \phi^{-1}(u) + \mu_2 \quad (17)$$

where,

$$\mu_k = \bar{X}_k, \sigma_k = S_{X_k}^*, (k = 1, 2). \quad (18)$$

Conclusively, the algorithm for implementing 2-D copula-EDA is as follows [35]:

1. Initialize (**pop**, N). Randomly generate initial population **pop** with size N . set generation count $g \leftarrow 0$.
2. Selection (**pop**, **spop**, *select-rate*). Select the best *select-rate* $\times N$ agents from **pop** to **spop** according to the agents' fitness.
3. copula-generator (**pop**, **spop**, *mutate-rate*).
 - 3.1. Construct the distribution model of **spop**:
 - 1) calculate the sample average and the sample variance for each random variable according to (13), then the marginal distributions are $F = N(\bar{X}_1, S_{X_1}^*)$ and $G = N(\bar{X}_2, S_{X_2}^*)$;
 - 2) calculate the estimation value of parameter ρ according to equation (11) or (12), then the copula C is the same as (14);
 - 3.2. Generate a new population by iterative using procedure generation(C, F, G), where

$$v = C_u^{(-1)}(\omega) = \sqrt{1 - \rho^2} \phi^{-1}(\omega) + \rho \phi^{-1}(u).$$
 - 3.3. Randomly generate some agents by the *mutate-rate*.
4. Stop if the termination criterion is met.
5. Set $g \leftarrow g+1$ and go to Step 2.

The following 9 test functions are used to show the behavior of the proposed algorithm cEDA and to compare the cEDA with PBILc and other EDAs. The test functions $F_1 \sim F_3$ and $F_6 \sim F_8$ are also used in [26].

- $F_1(x) = -\frac{100}{10^{-5} + \sum_i |y_i|}$, where $y_1 = x_1, y_i = x_i + y_{i-1} (i \geq 2), x_i \in [-3, 3]$, the optimal result is $F_1^*(0, 0, \dots, 0) = -10^7$.
- $F_2(x) = -\frac{100}{10^{-5} + \sum_i |y_i|}$, where $y_1 = x_1, y_i = x_i + \sin y_{i-1} (i \geq 2), x_i \in [-3, 3]$, the optimal result is $F_2^*(0, 0, \dots, 0) = -10^7$.
- $F_3(x) = -\frac{100}{10^{-5} + \sum_i |y_i|}$, where $y_i = 0.024 \times (i-1) - x_i, x_i \in [-3, 3]$, the optimal result is $F_3^*(0.024 \times 2, 0.024 \times 3, \dots, 0.024 \times (n+1)) = -10^7$.

- $F_4(x) = \sum_i x_i^2$, where $x_i \in [-500, 500]$, the optimal result is $F_4^*(0, 0, \dots, 0) = 0$.
- $F_5(x) = 1 + \sum_i (\sin x_i)^2 - 0.1 \exp(-\sum_i x_i^2)$, where $x_i \in [-10, 10]$, the optimal result is $F_5^*(0, 0, \dots, 0) = 0.9$.
- $F_6(x) = \sum_i (x_i^2 - A \cos(2\pi x_i) + A)$, where $x_i \in [-5, 5]$, the optimal result is $F_6^*(0, 0, \dots, 0) = 0$.
- $F_7(x) = \sum_i (418.9829 + x_i \sin \sqrt{|x_i|})$, where $x_i \in [-500, 500]$, the optimal result is $F_7^*(-420.9687, -420.9687, \dots, -420.9687) = 0$.
- $F_8(x) = \sum_i x_i^2 - \prod_i \cos(\frac{x_i}{\sqrt{i+1}})$, where $x_i \in [-100, 100]$, the optimal result is $F_8^*(0, 0, \dots, 0) = -1$.
- $F_9(x) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$, where $x_1, x_2 \in [-2, 2]$, the optimal result is $F_9^*(0, -1) = 3$.

All test functions are optimized in 2-dimensional spaces, the maximal generation g is set to 1000. The search terminates if the distance between the best solution found so far and the optimum is less than the predefined precision. Table 1 and Table 2 display the experimental results.

Table 1 The convergence of copula-EDA and PBILc, the maximal generation is 1000

Functions	Algorithm	Population size	Select rate	Mutate rate	precision	convergence rate	convergence generations
F_1	cEDA	500	0.2	0.05	10^{-5}	50/50	50.700
	PBILc	500	0.2	0	10^{-5}	50/50	405.660
F_2	cEDA	500	0.2	0.05	10^{-5}	46/50	43.369
	PBILc	500	0.2	0	10^{-5}	50/50	421.120
F_3	cEDA	500	0.2	0.05	10^{-5}	50/50	17.600
	PBILc	500	0.2	0	10^{-5}	50/50	396.420
F_4	cEDA	100	0.2	0.05	10^{-5}	50/50	28.791
	PBILc	100	0.2	0	10^{-5}	50/50	841.400
F_5	cEDA	100	0.2	0.05	10^{-5}	50/50	222.760
	PBILc	100	0.2	0	10^{-5}	50/50	560.860
F_6	cEDA	100	0.2	0.05	10^{-5}	50/50	76.800
	PBILc	100	0.2	0	10^{-5}	50/50	493.600
F_7	cEDA	100	0.2	0.05	10^{-3}	44/50	69.386
	PBILc	100	0.2	0	10^{-3}	32/50	905.593
F_8	cEDA	100	0.2	0.05	10^{-5}	50/50	25.250
	PBILc	100	0.2	0	10^{-5}	50/50	664.300
F_9	cEDA	100	0.2	0.05	10^{-5}	50/50	21.306
	PBILc	100	0.2	0	10^{-5}	41/50	644.195

Table 2 The state of different algorithm after 50 runs

	copula_EDA	PBILc	PBILcM
	Mean±Std	Mean±Std	Mean±Std
F_1	-9.7229e+6±1.5966e+4	-9.9473e+6±3.1487e+4	-4.4996e+5±4.4091e+5
F_2	-9.4549e+6±1.3826e+3	-9.9403e+6±3.1134e+4	-3.8927e+5±3.7505e+5
F_3	-9.8119e+6±1.1698e+3	-9.9519e+6±2.0524e+4	-9.4466e+4±6.6105e+4
F_4	3.6721e-12 ±2.6923e-12	7.8393e-12 ±8.4951e-12	0.2069 ± 0.8377
F_5	0.9000±3.0600e-11	0.9000 ±3.3826e-14	0.9000 ±2.0910e-5
F_6	8.3590e-14±5.7045e-14	2.2464e-12 ±2.6022e-12	5.8463e-4 ±6.0017e-4
F_7	0.1596±0.7787	35.1368 ±58.5155	0.3105 ± 0.2903
F_8	-1.0000±4.4546e-14	-1.0000 ±3.6992e-13	-0.9907 ± 0.0388
F_9	3.0000±9.9700e-9	7.8600 ±10.4784	3.0116 ± 0.0206

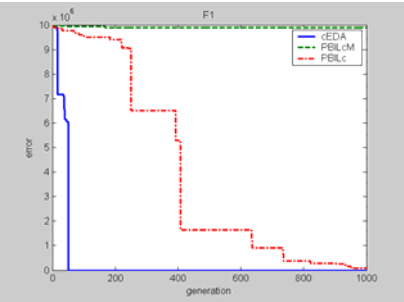


Fig. 3 The performance on test function F_1

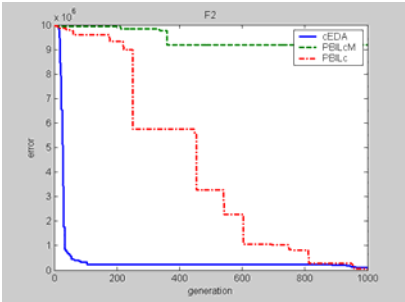


Fig. 4 The performance on test function F_2

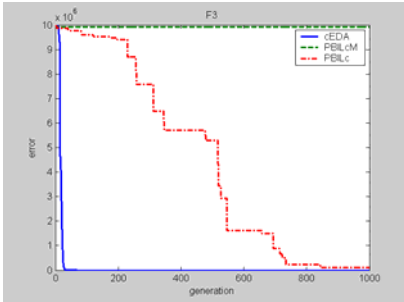


Fig. 5 The performance on test function F_3

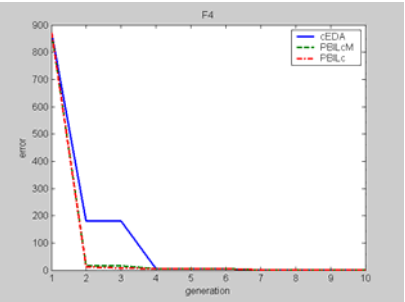


Fig. 6 The performance on test function F_4

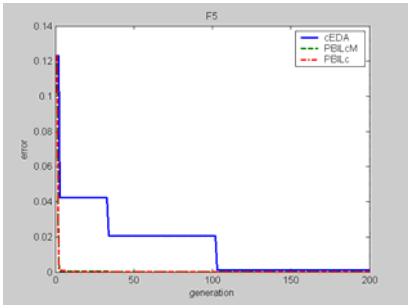


Fig. 7 The performance on test function F_5

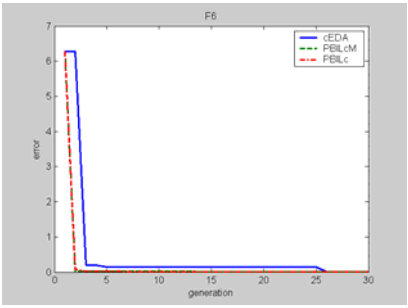


Fig. 8 The performance on test function F_6

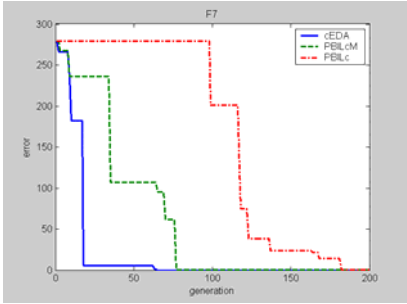


Fig. 9 The performance on test function F_7

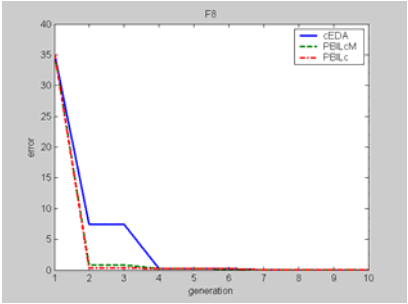


Fig. 10 The performance on test function F_8

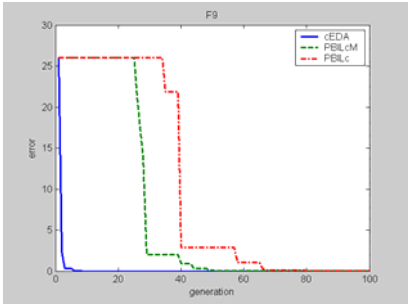


Fig. 11 The performance on test function F_9

Copula-EDA is abbreviated to cEDA in Table 1 and Fig.3-11. PBILcM is performed based on PBILc with the mutation rate as 0.05 for the sake of comparing the performance of copula-EDA and PBILc with the same parameters. But the experimental results show that the convergence rate of PBILcM is 0 in each function with the parameters in Table 1. The convergence rate and the convergence

generations are the average results of 50 runs. The experimental results show that copula-EDA converges to the global optimum quickly in the test functions.

The mean fitness and the standard variance of each algorithm after 50 runs are showed in Table 2. Obviously, copula-EDA performs better than the other two algorithms. It is found through further experiments that Copula-EDA converges faster than PBILc, but it needs more agents in the population than PBILc.

The evolution processes of the above three algorithms is compared with the same population in Fig.3-11. The population in the first generation is same for each algorithm. Error in Fig.3-11 is the fitness difference. Copula-EDA converges to the best solution quickly in almost all tested functions especially for the function F_1 - F_3 whose random variables are strongly correlative with each other. It is also can be seen from Fig.7-9 that copula-EDA is premature in some cases.

4.2 Archimedean Copula-EDAs

According to Sklar's theorem, two steps are performed in order to construct the joint probability distribution function of a random vector. The first step is constructing the margins of each random variable separately. The second step is selecting a proper copula to construct the joint distribution. Therefore, the distribution character of each random variable and their relationship can be studied by themselves. This way can be used in EDAs to model the joint probability distribution function. And then samples are generated from the specified joint distribution by use of the copula.

The optimization problem is

$$\min f(X) = f(x_1, x_2), \quad x_i \in [a_i, b_i] \quad (i=1, 2). \quad (19)$$

Denote the selected population with size s as

$$\mathbf{x} = \{x^i = (x_1^i, x_2^i), i=1, 2, \dots, s\} \quad (20)$$

In other words, \mathbf{x} are the s observations of the random vector (X_1, X_2) . The marginal distribution function of each random variable X_i can be estimated by normal distribution, t-distribution or empirical distribution, etc. Denote the marginal distribution function of X_i as $u=F(x_1)$ and $v=G(x_2)$. The joint probability distribution function is constructed with a selected copula C and the estimated margins in the light of Sklar's theorem.

The next step is to generate samples from the joint distribution using the copula as a tool. By virtue of Sklar's theorem, it is necessary only to generate a pair (u, v) of observations of uniform (0,1) random variables (U, V) whose joint distribution function is C , and then transform those uniform random variables via the quasi-inverse of the marginal distribution functions. One procedure for generating such of a pair (u, v) of uniform (0,1) random variables is the conditional distribution method. For this method, the conditional distribution function for V given $U = u$ is need, which is denoted as $C_u(v)$:

$$C_u(v) = P(V \leq v | U = u) = \lim_{\Delta u \rightarrow 0} \frac{C(u + \Delta u, v) - C(u, v)}{\Delta u} = \frac{\partial C(u, v)}{\partial u}. \quad (21)$$

$C_u(v)$ exists and is non-decreasing almost everywhere in \mathbf{I} .

Conclusively, the generation of sample is performed as the following steps:

1. Generate two independent uniform (0, 1) random variables u and t ;
2. Set $v = C_u^{(-1)}(t)$, where $C_u^{(-1)}(t)$ denotes a quasi-inverse of $C_u(v)$.
3. The desired pair is (u, v) .
4. Set $x_1 = F^{(-1)}(u)$, $x_2 = G^{(-1)}(v)$, then (x_1, x_2) is a sample of the specified joint distribution.

To sum up, the process for implementing 2-D Copula-EDA is as follows [36]:

1. Initialize (**pop**, N). Randomly generate initial population **pop** with size N . set generation count $g \leftarrow 0$.
2. Selection (**pop**, **spop**, *select-rate*). Select the best *select-rate* $\times N$ agents from **pop** to **spop** according to the agents' fitness.
3. Copula-generator (**pop**, **spop**, *mutate-rate*).
 - 3.1. Construct the distribution model of **spop**;
 - 3.2. Generate a new population based on the specified joint distribution, and randomly generate some agents by the rate *mutate-rate*.
4. Stop if the termination criterion is met.
5. Set $g \leftarrow g + 1$, and then go to step 2.

The functions in Sect. 4.1 are used to test the effectiveness of the proposed algorithm. The following two Archimedean copulas are chosen.

- Clayton: $C_1(u, v) = (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$, $\theta \geq -1, \theta \neq 0$
- Ali-Mikhail-Haq: $C_2(u, v) = \frac{uv}{1 - \theta(1-u)(1-v)}$, $-1 \leq \theta < 1$

All the one-dimensional marginal distributions are normal distributions. Table 3 displays the experimental results.

All test functions are optimized in 2-dimensional spaces, the maximal generation g is set to 1000. The search terminates if the distance between the best solution found so far and the optimum is less than the predefined precision (10^{-5} for other test functions in spite of 10^{-3} for F_7). Parameters are set to (*select-rate*=0.2, *mutate-rate*=0.05, population size $N=100$) for all experiments. The convergence rate and the convergence generations are the average results of 50 runs. The experimental results show that Copula-EDA converges to the global optimum quickly in the test functions. There is not much difference in performance between two copulas for other test functions despite F_3 and F_6 . Both the algorithms proposed in this section perform better than the copula-EDA based on Gaussian copula (see Sect. 4.1) and PBILc [26].

Table 3 Experimental results of Archimedean copula-EDAs

Test Function	Copula	Convergence Rate	Convergence Generation
F_1	C_1	50/50	13.7400
	C_2	50/50	13.0600
F_2	C_1	50/50	13.6800
	C_2	50/50	13.1400
F_3	C_1	50/50	4.8000
	C_2	50/50	11.7200
F_4	C_1	50/50	16.2800
	C_2	50/50	16.7000
F_5	C_1	50/50	75.7800
	C_2	50/50	76.7800
F_6	C_1	50/50	43.6800
	C_2	50/50	28.7400
F_7	C_1	47/50	55.5957
	C_2	50/50	32.8600
F_8	C_1	50/50	14.9600
	C_2	50/50	15.1200
F_9	C_1	48/50	12.2083
	C_2	50/50	12.2600

5 High-Dimensional Copula-EDAs

In fact, many optimization problems are high-dimensional. The 2-dimensional copula-EDAs only show the feasibility to apply copula theory to EDAs. The copula-EDA based on the empirical copula is discussed in this section.

5.1 High-Dimensional Copula Constructions

Let $F(\mathbf{x})$ be the joint of the D -dimensional random vector \mathbf{x} , the margins of each random variable are $F_i(x)(i=1,2,...,D)$, and the copula, then the equation $C(u_1,u_2,...,u_D)=F(F_1^{-1}(u_1), F_2^{-1}(u_2)...., F_D^{-1}(u_D))$ is gotten according to Sklar's theorem. A random vector $(u_1,u_2,...,u_D)$ obeying $C(\mathbf{u})$ is generated firstly in order to generate a random vector obeying $F(\mathbf{x})$. Subsequently, the vector $(x_1,x_2,...,x_D)$ is

calculated according to the inverse function of the margins $F_i(x)(i=1,2,\dots,D)$, i.e. $x_i = F_i^{-1}(x)(i=1,2,\dots,D)$.

If the density of the copula $C(\mathbf{u})$ is $c(\mathbf{u})$, then the marginal probability density functions are

$$c_d(u) = c_d(u_1, u_2, \dots, u_d) = \int_{u_{d+1}=0}^1 \dots \int_{u_D=0}^1 c(u) du_{d+1} \dots du_D, d = 1, 2, \dots, D-1 \quad (22)$$

Obviously, $c(\mathbf{u}) = c_D(\mathbf{u})$. The condition distribution functions are

$$C_d(u_d | u_1, \dots, u_{d-1}) = P\{U_d \leq u_d | U_1 = u_1, \dots, U_{d-1} = u_{d-1}\} = \frac{\int_{u=0}^{u_d} c_d(u_1, \dots, u_{d-1}, u) du}{c_{d-1}(u_1, \dots, u_{d-1})}, d=1, 2, \dots, D \quad (23)$$

A method to construct a copula based on the empirical distribution and to sample from the empirical copula is proposed in [34] and it is briefly introduced in the following paragraphs.

Let the samples of the D -dimensional random vector \mathbf{z} be $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iD})(i=1, 2, \dots, n)$. The empirical distribution function of each dimension is constructed firstly according to the samples and then each sample \mathbf{z}_i is mapped to a value u_i belonging to \mathbf{I}^D . The values $\mathbf{u}_i (i=1, 2, \dots, n)$ are actually the samples of the random vector \mathbf{u} obeying cdf $C(\mathbf{u})$. In order to sample from the estimated cdf $C(\mathbf{u})$, the conditional distribution functions $c_d(\mathbf{u})$ are necessary to be estimated according to the samples $u_i (i=1, 2, \dots, n)$. the details are shown as follows.

The samples $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iD})(i=1, 2, \dots, n)$ are sorted in each dimension. For example, the sorted sequence of the d^{th} dimension is $z_{(1)d}, z_{(2)d}, \dots, z_{(n)d}$, then the empirical distribution of the d^{th} dimension is

$$F_d(z) = \begin{cases} 0 & z < z_{(1)d} \\ i/n & z_{(i)d} \leq z \leq z_{(i+1)d}, i = 1, 2, \dots, n-1 \\ 1 & z_{(n)d} \leq z \end{cases} \quad (24)$$

According to equation (24), the samples $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{iD})(i=1, 2, \dots, n)$ are mapped to the vectors $\mathbf{u}_i = (u_{i1}, u_{i2}, \dots, u_{iD}) \in \{1/n, 2/n, \dots, 1\}^D (i=1, 2, \dots, n)$.

Denote S_1, S_2, \dots, S_K as the partition of the interval $\mathbf{I}=[0,1]$, where $S_i = ((i-1)\delta, i\delta] (i=1, 2, \dots, K)$, $\delta=1/K$, K is a positive integer. Thus the D -cube \mathbf{I}^D is divided into K^D subcubes $\Delta_i = S_{i_1} \times S_{i_2} \times \dots \times S_{i_D}$, $\mathbf{i} = (i_1, i_2, \dots, i_D) \in \{1, 2, \dots, K\}^D$. denote N_i as the number of points in Δ_i , i.e. $N_i = |\{\mathbf{u}_j | \mathbf{u}_j \in \Delta_i\}|$. The density of the copula is

$$c(\mathbf{u}) = f_i = N_i / n / \delta^D \quad (25)$$

Let $\uparrow u = \max\{1, \lceil uK \rceil\}$, then $\mathbf{i} = (\uparrow u_1, \uparrow u_2, \dots, \uparrow u_D) \in \{1, 2, \dots, K\}^D$, and

$$c_d(u_1, \dots, u_d) = \delta^{D-d} f_{\uparrow u_1, \dots, \uparrow u_d}^{(d)}, d = 1, \dots, D-1 \quad (26)$$

where,

$$f_{i_1, \dots, i_d}^{(d)} = \sum_{(i_{d+1}, \dots, i_D) = (1, \dots, 1)}^{(K, \dots, K)} f_i, \quad (i_1, \dots, i_d) = \{1, 2, \dots, K\}^d \quad (27)$$

The conditional distribution is

$$C_2(u_2 | u_1) = \delta^{D-1} \sum_{i_2=1}^{\downarrow u_2} f_{\uparrow u_1, i_2}^{(2)} + (u_2 - \downarrow u_2 \delta) \delta^{D-2} f_{\uparrow u_1, \uparrow u_2}^{(2)} \quad (28)$$

$$C_d(u_d | u_1, \dots, u_{d-1}) = \frac{\delta \sum_{i_d=1}^{\downarrow u_d} f_{\uparrow u_1, \dots, \uparrow u_{d-1}, i_d}^{(d)} + (u_d - \downarrow u_d \delta) f_{\uparrow u_1, \dots, \uparrow u_d}^{(d)}}{\delta f_{\uparrow u_1, \dots, \uparrow u_{d-1}}^{(d-1)}} \quad (29)$$

where, $\downarrow u = \uparrow u - 1$.

The following algorithm (Algorithm 1) can be applied to calculate the arrays $f^{(d)}$ which we use for the calculation of the conditional distribution functions of the copula.

1. Calculate the empirical distribution functions of the marginal distributions of the sample with equation (24) and the image points \mathbf{u}_i of the sample points $\mathbf{z}_i (i=1, 2, \dots, n)$.
2. Set all elements of the arrays $f^{(d)}$ to 0.
3. for $i:=1$ to n do

for $d:=1$ to D do

$j_d := \uparrow u_{d,i}$;

end

for $d:=2$ to D do

$f_{j_1, \dots, j_d}^{(d)} := f_{j_1, \dots, j_d}^{(d)} + 1/n / \delta^D$

end

end

Let $\mathbf{f} = \{f_i^d\}$, $(\mathbf{i} = (i_1, i_2, \dots, i_D) \in \{1, 2, \dots, K\}^D, d=1, 2, \dots, D-1)$. The procedure *generation*(\mathbf{f}) can be used to generate random numbers u_1, u_2, \dots, u_D obeying the empirical copula.

1. Generate two independent random numbers u_1 and u ;

2. Calculate $u_2 = \downarrow u_2 \delta + \frac{u - \delta^{D-1} \sum_{i_2=1}^{\downarrow u_2} f_{\uparrow u_1, i_2}^{(2)}}{\delta^{D-2} f_{\uparrow u_1, \uparrow u_2}^{(2)}}$, where $\downarrow u_2$ is the minimal number in

$\{0, 1, \dots, K-1\}$ satisfied with the condition $uK^{D-1} \leq \sum_{i_2=1}^{\downarrow u_2+1} f_{\uparrow u_1, i_2}^{(2)}$;

3. for $d=3$ to D

– 3.1. Generate a random number u in \mathbf{I} ;

– 3.2. Calculate $u_d = \downarrow u_d \delta + \delta \frac{u f_{\uparrow u_1, \dots, \uparrow u_{d-1}}^{(d-1)} - \sum_{i_d=1}^{\downarrow u_d} f_{\uparrow u_1, \dots, \uparrow u_{d-1}, i_d}^{(d)}}{f_{\uparrow u_1, \dots, \uparrow u_d}^{(d)}}$, where $\downarrow u_d$ is the minimal number in $\{0, 1, \dots, K-1\}$ satisfied with the condition $u f_{\uparrow u_1, \dots, \uparrow u_{d-1}}^{(d-1)} \leq \sum_{i_d=1}^{\downarrow u_d+1} f_{\uparrow u_1, \dots, \uparrow u_{d-1}, i_d}^{(d)}$

end

5.2 Copula-EDA Based on Empirical Copula

For the optimization problem $\min f(x_1, x_2, \dots, x_D)$, $x_i \in [a_i, b_i], (i=1, 2, \dots, D)$, the copula-EDA based on the above empirical copula is in the following [37].

1. Generate N individuals in the search space randomly. Decide the selection rate s and the mutation rate t ;
2. Select $n=s \times N$ individuals $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$ among the N individuals as the promising population, and reserve them in the next generation;
3. Calculate $\mathbf{f}=\mathbf{F}(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n)$ by doing Algorithm 1
4. Repeat the following steps $m=(1-s-t) \times N$ times, add the new m individuals into the population.
 - 4.1 $(u_1, u_2, \dots, u_D) = \text{generation}(\mathbf{f})$;
 - 4.2 for $i:=1$ to D

$$x_i = F_i^{-1}(u_i), \text{ where } F_i \text{ is the empirical distribution (24) or the normal distribution}$$
 - end
 - (x_1, x_2, \dots, x_D) is the new individual.
5. Generate $t \times N$ individuals in the search space randomly as the new individuals;
6. Stop if the terminate condition is met, and the best individual of the population is the optimal result.

The following 3 functions are used to test the proposed algorithm.

- Sumcan function: $f_1(x) = -\{10^{-5} + \sum_{i=1}^D |y_i|\}^{-1}$, where, $y_1=x_1$, $y_i=y_{i-1}+x_i$, $i=2, \dots, d$, $-0.16 \leq x_i \leq 0.16$
- Schwefel function: $f_2(x) = \sum_{i=1}^D [(x_1 - x_i^2)^2 + (x_i - 1)^2]$, where, $-10 \leq x_i \leq 10$

- Griewank function: $f_3(x) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})$, where, $-600 \leq x_i \leq 600$

The above three functions are all minimal optimization problem. The empirical copula and the normal margins are used in the tested copula-EDA, and is denoted as cEDAPG. The parameters are the same as in [30], *i.e.* the dimension of the problem is D=10, the population size of the three problem are 2000, 2000 and 750 respectively. The maximal fitness evaluation is 300,000. The truncation selection is used. The mutation is used and the mutation rate is 0.05. The search space is not divided when the empirical copula is constructed, *i.e.* K=1. The variance for sampling is one of the following three strategies.

A. The variance is fixed.

B. The variance is set to the sample variance, *i.e.* $\sigma_i = \sqrt{\frac{\sum_{j=1}^S (x_{ij} - \bar{x}_i)^2}{S-1}}$,

where, $\bar{x}_i = \frac{1}{S} \sum_{j=1}^S x_{ij}$.

C. The variance is linearly changed, *i.e.*

$$\sigma_i^{t+1} = (1-\alpha)\sigma_i^t + \alpha \sqrt{\frac{\sum_{j=1}^S (x_{ij} - \bar{x}_i)^2}{S-1}}, \alpha=0.2$$

Table 4 The comparison among cEDA and other algorithms on Sumcan function

Algorithm	Selection rate	σ	F1			
			mean	StdVar	min	max
UMDA _c ^G			-53460			
MIMIC _c ^G			-58775			
EGNA _{ec}			-100000			
EGNA _{BGe}			-100000			
ES			-5910			
cEDAPG	0.5	A:0.02	-30.2775	3.4345	-37.9727	-26.2.83
	0.5	A:0.05	-12.5174	1.4837	-15.7628	-11.0210
	0.2B	B	-60770	27564	-99866	-6390.9
	0.2	B	-1074.6	482.2301	-2168.2	-513.8331
	0.5	C	-184.0976	52.9640	-235.0931	-46.0561
	1/3	C	-561.7312	178.8254	-764.9960	-114.3033
	0.2B	C	-9910.3	2692.9	-12511	-2786.9
	0.2	C	-2089.4	587.4801	-2784.2	-547.5289

Table 5 The comparison among cEDA and other algorithms on Schwefel function

Algorithm	Selection rate	σ	F2			
			mean	StdVar	min	max
UMDA _c ^G MIMIC _c ^G EGNA _{ee} EGNA _{BGe} ES			0.13754 0.13397 0.09914 0.0250 0			
cEDAPG	0.5	A::02	4.0743	0.8572	2.8008	5.3325
	0.5	A::05	0.0097	0.0032	0.0061	0.0167
	0.2B	B	2.3833×10^{-4}	5.2429×10^{-5}	1.2190×10^{-4}	3.6281×10^{-4}
	0.2	B	0.1478	0.0259	0.1066	0.1954
	0.5	C	8.2443×10^{-4}	0.0019	1.0471×10^{-4}	0.0063
	1/3	C	1.8707×10^{-5}	4.6817×10^{-5}	2.8442×10^{-6}	1.5194×10^{-4}
	0.2	C	3.0034×10^{-7}	7.3179×10^{-7}	4.0579×10^{-8}	2.3825×10^{-6}

Table 6 The comparison among cEDA and other algorithms on Griewank function

Algorithm	Selection rate	σ	F3			
			mean	StdVar	min	max
UMDA _c ^G MIMIC _c ^G EGNA _{ee} EGNA _{BGe} ES			0.011076 0.007794 0.008175 0.012605 0.034477			
cEDAPG	0.5	A::02	1.0796	0.1977	0.6106	1.2775
	0.5	A::05	0.5489	0.3694	0.2215	1.4507
	0.2B	B	0.0046	0.0151	0	0.0857
	0.2	B	0	0	0	0
	0.5	C	1.3192×10^{-9}	3.1529×10^{-9}	1.2085×10^{-10}	1.0288×10^{-8}
	1/3	C	4.2966×10^{-14}	1.0565×10^{-13}	7.2164×10^{-15}	3.4361×10^{-13}
	0.2	C	0	0	0	0

The experimental results in Table 4-6 indicate that the performance of cEDAPG is affected by the parameters. cEDA finds the better results than other compared algorithms in certain parameters. Especially for Griewank function, cEDAPG finds the optimal result after 68.7 generations when selection rate is 0.2 and the variance is changed in strategy B. It can be indicated from Fig. 12-13 that cEDAPG is good at exploration and is not good at exploitation.

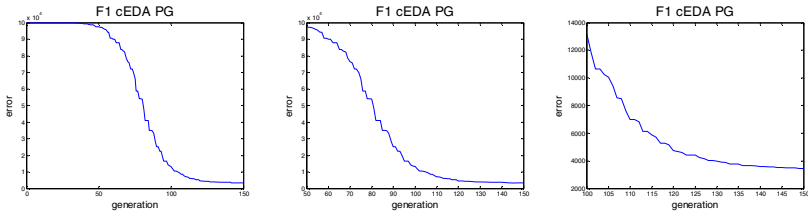


Fig. 12 The performance of cEDA_{PG} in Sumcan function

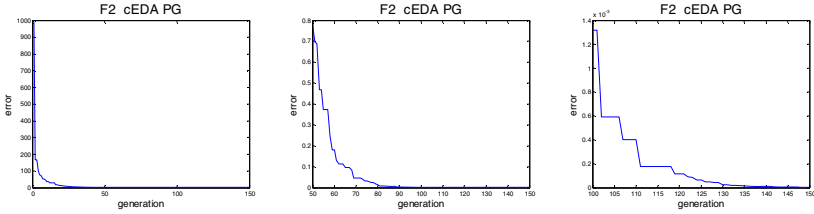


Fig. 13 The performance of cEDA_{PG} in Schwefel function

6 Conclusion

Estimation of Distribution Algorithms utilize the distribution characters of the selected population to guide the creation of the next generation. Therefore, the population evolves towards the optimal direction and finds the optimal solution quickly. Two key steps in EDAs are to estimate the distribution of the selected population and to sample from the distribution model.

Copula theory contributes to the estimation of the distribution model and sampling from the estimated model because copula theory provides a way to present the multivariate joint distribution with the univariate marginal distributions and a function called copula. How to construct copula and to sample from the copula are also the research topics of copula theory.

Copula theory is applied into the research of EDAs and three EDAs based on different copulas are introduced in this chapter. They are respectively copula-EDA based on Gaussian copula, copula-EDA based on Archimedean copula and copula-EDA based on empirical copula. All of them can find the optimal results quick than the other algorithms used for comparison. But there are also shortages in them. For example, the copula-EDA based on empirical copula is not good at exploitation and the convergence speed in last generations is slow and the algorithm is affected by the selection of parameters.

Different copula is corresponded to different relationship of the random variables. So the performance of copula-EDAs based on different copula is different on optimization problems. The following questions need to be studied further.

- The expansion of copula-EDA based on Gaussian copula and Archimedean copula on high-dimensional optimization problems.
- How to select and construct copula.

- The comparison among the copula-EDAs based on different copulas.
- The relationship between the characters of the optimization problem and the optimal copula.

References

- [1] Larranaga, P., Lozano, J.A.: *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston (2002)
- [2] Baluja, S.: *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. Citeseer (1994), doi: 10.1.1.9.8084
- [3] Baluja, S.: *An Empirical Comparison of Seven Iterative and Evolutionary Function Optimization Heuristics*. Technical Report CMU-CS-95-193, Computer Science Department, Carnegie Mellon University (1995)
- [4] Hohfeld, M., Rudolph, G.: *Towards a Theory of Population Based Incremental Learning*. In: *Proc. of the 4th International Conference on Evolutionary Computation*, pp. 1–5 (1997)
- [5] Cristina, G., Lozano, J.A., Larranaga, P.: *Analyzing the PBIL Algorithm by means of Discrete Dynamical Systems*. *Complex Syst.* 12(4), 465–479 (2001)
- [6] 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)
- [7] Muhlenbein, H.: *The Equation for Response to Selection and its use for Prediction*. *Evol. Comput.* 5(3), 303–346 (1997)
- [8] Shapiro, J.L.: *Drift and Scaling in Estimation of Distribution Algorithms*. *Evol. Comput.* 13(1), 99–123 (2005)
- [9] Harik, G.R., Lobo, F.G., Goldberg, D.E.: *The compact genetic algorithm*. In: *Proc. IEEE Conf. Evol Computation*, Indianapolis, USA, pp. 523–528 (1998)
- [10] De Bonet, J.S., Isbell, C.L., Viola, P.: *MIMIC: Finding optima by estimation probability densities*. In: Becker, S. (ed.) *Advances in Neural Information Processing Systems*, pp. 424–430. MIT Press, Cambridge (1997)
- [11] Baluja, S., Davies, S.: *Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space*. In: *Proc. of the 14th international conference on machine learning*, pp. 30–38. Morgan Kaufmann, San Francisco (1977)
- [12] Pelican, M., Muhlenbein, H.: *The birandom variable marginal distribution algorithm*. In: Furuhashi, T. (ed.) *Advances in Soft Computing-Engineering Design and Manufacturing*, pp. 521–535. Springer, London (1999)
- [13] Pelikan, M., Muhlenbein, H.: *Marginal Distributions in Evolutionary Algorithms*. In: *Proc. of the International Conference on Genetic Algorithms*, pp. 90–95. Technical University of Brno, Brno (1998)
- [14] Harik, G.: *Linkage Learning via Probabilistic Modeling in the ECGA*, Illigal Rep. No.99010, Illinois Genetic Algorithms Lab., University of Illinois, Urbana-Champaign, Illinois (1999)
- [15] Muhlenbein, H., Mahnig, T.: *FDA- a scalable evolutionary algorithm for the optimization of additively decomposed functions*. *Evolu. Comput.* 7(4), 353–376 (1999)

- [16] Muhlenbein, H., Mahnig, T.: Convergence Theory and Applications of the Factorized Distribution Algorithm. *Journal of Comput and Information Technology* 7(1), 19–32 (1999)
- [17] Muhlenbein, H., Mahnig, T.: The Factorized Distribution Algorithm for Additively Decomposable Functions. In: *Second Symposium on Artificial Intelligence. Adaptive Systems. CIMA F 1999, La Habana*, pp. 301–313 (1999)
- [18] Muhlenbein, H., Mahnig, T.: Evolutionary algorithms: From recombination to search distributions. In: Kallel, L., Naudts, B., Rogers, A. (eds.) *Theoretical Aspects of Evolutionary Computing.*, pp. 137–176. Springer, London (2001)
- [19] Muhlenbein, H., Mahnig, T., Ochoa, A. R.: Distributions and Graphical Models in Evolutionary Optimization. *J. Heuristics* 5, 215–247 (1999)
- [20] 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)
- [21] Pelikan M.: A Simple Implementation of Bayesian Optimization Algorithms in C++(Version 1.0). *Illegal Report No. 99011*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois (1999)
- [22] Pelikan, M., Goldberg, D.E., Cantu-Paz, E.: Bayesian Optimization Algorithm, Population Sizing, and Time to Convergence. *IlleGAL Report No. 2000001*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana Champaign, Urbanan, Illinois (2000)
- [23] Pelikan, M.: *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*. Springer, New York (2005)
- [24] Gao, Y., Culberson, J.: Space Complexity of Estimation of Distribution Algorithms. *Evol. Comput.* 13(1), 125–143 (2005)
- [25] Pelikan, M., Sastry, K., Goldberg, D. E.: Scalability of the Bayesian Optimization Algorithm. *International Journal of Approximate Reasoning* 31(3), 221–258 (2002)
- [26] 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)
- [27] Larranaga, P., Etxeberria, R., Lozano, J.A., Pena, J.M.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: *Proc. 2000 Genetic and Evolutionary Computation Conf. Workshop Program, Las Vegas, Nevada*, pp. 201–204 (2000)
- [28] Rudlof, S., Koppen, M.: Stochastic Hill Climbing by Vectors of Normal Distributions. In: *Proc. of the first online workshop on soft computing(WSC1)*, Nagoya, Japan (1996)
- [29] Larrañaga, P., Lozano, J.A., Bengoetxea, E.: Estimation of Distribution Algorithms based on multivariate normal and Gaussian networks. *Technical Report KZZA-IK-1-01 of the Department of Computer Science and Artificial Intelligence, University of the Basque Country, Spain* (2001)
- [30] Larranaga, P., Etxeberria, R., Lozano, J.A., Pena, J.M.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: *Proc. of the Genetic and Evolutionary Computation Conference, GECCO 2000, Las Vegas, Nevada, USA, July 8-12*, pp. 201–204. Morgan Kaufmann, San Francisco (2000)
- [31] Bosman, P.A.N., Thierena, D.: Expanding from Discrete to Continuous Estimation of Distribution Algorithms: the IDEA. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) *PPSN 2000. LNCS*, vol. 1917, pp. 767–776. Springer, Heidelberg (2000)

- [32] Nelsen, R.B.: An introduction to copulas, 2nd edn. Springer, New York (2006)
- [33] Whelan, N.: Sampling from Archimedean Copulas (2004),
<http://home.golden.net/~annaw/Niall/papers/Archimedean.pdf>
(Accessed January 1, 2009)
- [34] Strelan, J.C., Nassaj, F.: Analysis and generation of random vectors with copulas. In: Henderson, S.G., Biller, B., Hsieh, M.H., Shortle, J., Tew, J.D., Barton, R.R. (eds.) Proc. of the 2007 Winter Simulation Conference (WSC 2007), Washington, DC, USA, pp. 488–496 (2007)
- [35] Wang, L.F., Zeng, J.C., Hong, Y.: Estimation of Distribution Based on Copula Theory. In: IEEE CEC 2009, Trondheim, Norway, May 18–21, pp. 1057–1063 (2009)
- [36] Wang, L.F., Zeng, J.C., Hong, Y.: Estimation of Distribution Based on Archimedean Copulas. In: ACM SIGEVO GEC 2009, Shaihai, China, June 12–14, pp. 993–996 (2009)
- [37] Wang, L.F., Zeng, J.C., Hong, Y.: Estimation of Distribution Algorithm Modeling and Sampling by means of Copula. Submitted to: Chinese journal of computers (in Chinese) (2009)