Estimation of Distribution Algorithm Based on Copula Theory

Conference Paper · May 2009

DOI: 10.1109/CEC.2009.4983063 · Source: DBLP

Jian-Chao Zeng
Taiyuan University of Science and Technology

Estimation of Distribution Algorithm Based on Copula Theory

Li-Fang Wang, Jian-Chao Zeng, and Yi Hong

Abstract*:* Estimation of Distribution Algorithm (EDA) is a novel evolutionary computation, which mainly depends on learning and sampling mechanisms to manipulate the evolutionary search, and has been proved a potential technique for complex problems. However, EDA generally spend too much time on the learning about the probability distribution of the promising individuals. The paper propose an improved EDA based on copula theory (copula-EDA) to enhance the learning efficiency, which models and samples the joint probability function by selecting a proper copula and learning the marginal probability distributions of the promising population. The simulating results prove the approach is easy to implement and is validated on several problems.

I. Introduction

Estimation of Distribution Algorithm (EDA) has caught many researchers' eyes since proposed in 1996[1]. There are two important steps in EDAs which are explicitly modeling the probability distribution of the good solutions found so far and sampling new solutions by use of the constructed model. Many kinds of EDAs are proposed such as first order EDAs (PBIL[2], UMDA[3], cGA[4]) who take the variables as independent with each other, second order EDAs (MIMIC[5], KFMIMIC[6], BMDA[7]) who only consider the relationship between pairs of variables, high order EDAs(ECGA[8], FDA[9], BOA[10]) and EDAs in continuous search spaces(PBILc[11], UMDAc[12], PPCA[13], ENGA[14]).

EDAs are extended into continuous optimization problems recently. Gaussian distribution $N(\mu,\sigma)$ is used mainly for the estimation of the selected population in those continuous EDAs. The ways of setting parameters of Gaussian distribution are approximately the following.

1. The combination of the sample average or the sample variance [11], [13], [15], [16];

2. Linearly changing the parameters according to the distribution of the selected population [11], [17];

3. Using a constant value [11];

4. Using a self-adjust way [11], [18].

It has been proved in theory that EDAs are global converged if the population size is infinite and the constructed probability model can reflect exactly the probability distribution of the selected population, no matter which selection operator is used such as proportional selection, truncation selection or 2-agents tournament selection [19], [20]. However, both infinite population size and exact probability model are hard to realize, especially when the optimization problem is very large. For example, the algorithms proposed in [10] and [15] need to find the optimal network, and the finding work for a given dataset is a NP-complete problem. In other words, enormous time is required for the modeling work when the optimization problem size is large.

Copula theory proposed in 1960's [21] is developed quickly until recent years. The research productions on copula theory are available mainly in statistics [21]-[23], [26] and finance [24], [25] at present, we cannot find its available results in evolutionary computations. According to copula theory, a joint probability distribution can be decomposed into *n* marginal probability distributions and a copula function. So the marginal distributions and their dependency can be studied separately. The probability distribution of the selected population will be modeled in EDAs. Some available EDAs considered the variables are independent; others considered the relations between variables which need more time and spaces. Furthermore, the spent on learning the probability distribution increases quickly as the problem is more and more complex. It is easier to model the probability distribution of univariate compared with the joint probability distribution of multivariate. According to copula theory, the joint probability distribution of multivariate can be constructed utilizing a copula function and the marginal probability distributions of every variable. Therefore, an Estimation of Distribution Algorithms based on copula theory (copula-EDA) is proposed in this paper. This algorithm learns the probability distribution of promising population and samples agents on the ground of copula theory. This algorithm is simple and easy to realize.

The paper is organized as follows. In section II, the copula theory is introduced briefly firstly, and then the copula-EDA is produced, finally 2-D Gaussian copula-EDA is implemented as an example. In section III, the experimental results are presented. The conclusions are provided in section IV.

II. copula-EDA

A. A brief introduction to copula theory

Researchers have not attached the importance to copula theory when it was proposed. The global economy developed quickly in recent years, and the financial risk analysis become very important. As one of the useful tools of the financial risk analysis, copula theory arouses the interest of global researchers. The copula theory is briefly introduced in the following, the details can be found in [21].

1)Definition 1: If a two-dimensional function $C:[0,1]^2 \to [0,1]$, meet

1. $\forall t \in [0,1]$， then $C(t,0) = C(0,t) = 0$；

2. $\forall t \in [0,1]$， then $C(t,1) = C(1,t) = 1$；

3. $\forall u_1, u_2, v_1, v_2 \in [0,1]$ ， and $u_1 \le u_2, v_1 \le v_2$ ， then $C(u_2,v_2) - C(u_2,v_1) - C(u_1,v_2) + C(u_1,v_1) \ge 0$，

the function C is called two-dimensional copula or 2-copula。

2) Sklar's theorem: Let $H(x,y)$ be a joint distribution function with margins $F(x)$ and $G(y)$. Then there exists a copula C such that for all (x,y) in ***R***,

$$H(x,y) = C(F(x),G(y)) \tag{1}$$

If $F(x)$ and $G(y)$ are continuous, then C is unique; otherwise, C is uniquely determined on Ran*F* × Ran*G*. Conversely, if C is a copula and $F(x)$ and $G(y)$ are distribution functions, then the function $H(x,y)$ defined by (1) is a joint distribution function with margins $F(x)$ and $G(y)$.

Sklar's theorem is the foundation of copula theory. It elucidates the role that copulas play in the relationship between multivariate distribution functions and their univariate margins[21].

3) The classes of copulas: There are two classes of copulas: ellipse copulas and Archimedean copulas. Ellipse copulas include Gauss-copula and t-copula. Archimedean copulas are produced from different generators according to the definition of Archimedean copula.

Gaussian copula: C_R^{Ga} is a Gaussian copula if the random vector (X,Y) meets the following conditions:

1. The marginal distributions u and v obey Gaussian distribution;

2. The copula to connect the marginal distributions is also normal, i.e.

$$C_R^{Ga}(u,v) = \phi_R(\phi^{-1}(u), \phi^{-1}(v))$$

Where, ϕ_R is a multivariate standardized normal distribution whose correlation matrix is ***R***, its inverse function is ϕ^{-1}.

t-copula：Random vector (X,Y) obeys t-distribution whose freedom is γ, and the marginal distribution of each random variable is u and v separately, $C_{\gamma,R}{}^{t}$ is a t-copula if

$$C_{\gamma,R}{}^{t}(u,v) = t^n_{\gamma,R}(t_\gamma^{-1}(u), t_\gamma^{-1}(v)),$$

where, t_γ^{-1} denotes the inverse function of univariate standard t-distribution whose freedom is γ, $t^n_{\gamma,R}$ denotes n-dimensional standard t-distribution whose freedom is γ and linear correlation matrix is ***R***.

Archimedean copulas: Copulas of the form

$$C(u,v) = \phi^{[-1]}(\phi(u) + \phi(v))$$

are called Archimedean copulas, where, ϕ is a continuous, strictly decreasing function from [0,1] to [0, ∞] such that $\phi(1) = 0$, and $\phi^{[-1]}$ is the pseudo-inverse of ϕ defined by

$$\phi^{[-1]}(t) = \begin{cases} \phi^{-1}(t) & 0 \le t \le \phi(0) \\ 0 & \phi(0) \le t < \infty \end{cases}.$$

The function ϕ is called a generator of the copula.

B. The frame of copula-EDA

Sklar's theorem plays an important role in copula theory. It is got from Sklar's theorem that copula correlates the marginal distribution functions and the joint distribution function. If the marginal distribution functions and a copula are available, then the joint distribution function is easy to construct. In EDAs, the distribution of promising population must be estimated in every generation. In the light of copula theory, only the marginal distributions are needed to be estimated. The joint distribution function can be constructed by the marginal distribution functions and a well-selected copula. The next generation can be sampled using the joint distribution function.

Suppose that the optimization problem is

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

Denote the promising population with size s as $\mathbf{x} \triangleq \{x^i = (x^i{}_1, x^i{}_2, ..., x^i{}_n), i = 1,2,...,s)\}$, then $\mathbf{x}$ is the s observed values of random vector （$X_1, X_2, ..., X_n$）. The marginal distribution of each random variable $X_i \quad (i = 1,2,..n)$ can be Gaussian distribution, t-distribution or some others. On the ground of Sklar's theorem, the joint distribution function can be constructed with a selected copula and the marginal distributions. The modeling of the distribution of the selected population is finished.

Next step is to generate new population by Monte Carlo method [21], [25]. The definition of quasi-inverse is introduced firstly.

Definition 2 [21]: Let *F* be a distribution function. Then a quasi-inverse of *F* is any function $F^{(-1)}$ with domain [0,1] such that

1. If t is in Ran*F*, then $F^{(-1)}(t)$ is any number *x* in ***R*** such that $F(x) = t$, i.e., for all t in Ran*F*, $F(F^{(-1)}(t)) = t$;

2. If t is not in Ran*F*, then

$$F^{(-1)}(t) = \inf\{x \mid F(x) \ge t\} = \sup\{x \mid F(x) \le t\}.$$

2-copula is considered in the following, in other words, the random samples of two-dimensional random vector (X_1, X_2) will be produced. Given that the marginal distribution functions are $u = F(x_1)$ and $v = G(x_2)$, obviously $u \sim U[0,1]$ and $v \sim U[0,1]$, then the random vector sequences can be sampled from u and v . According to the properties of copulas, the condition distribution function $C_u(v) = \partial C(u,v) / \partial u$ is monotony non-decreasing in $v \in [0,1]$ when $u \in [0,1]$ is given. Denote $\omega \triangleq C_u(v) = \partial C(u,v) / \partial u$, then ω is uniform in interval [0,1]. Thus, generate u and ω by Monte Carlo method in interval [0,1] firstly. And then get $v = C_u^{(-1)}(\omega)$ from the quasi-inverse function of $C_u(v)$. Finally, from $x_1 = F^{(-1)}(u)$ and $x_2 = G^{(-1)}(v)$, the random vector (x_1, x_2) is obtained. The pseudo code of the generating process is showed in Fig. 1.

```
generation(C, F, G)
{
  --Generate two independent uniform (0,1) variates $u$ and $\omega$;
  --Set $v = C_u^{(-1)}(\omega)$, where $C_u^{(-1)}$ denotes a quasi-inverse of $C_u$;
  --set $x_1 = F^{(-1)}(u)$ and $x_2 = G^{(-1)}(v)$;
  --The desired pair is $(x_1, x_2)$.
}
```

Fig. 1. The pseudo code of generating an agent

C. *Two dimensional Gaussian copula-EDA*

A two-dimensional optimization problem is considered. Accordind to copula theory, the marginal distribution of each 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

$$C(u, v\ ;\rho) = \phi_\rho(\phi^{-1}(u), \phi^{-1}(v)), u, v \in [0,1], \tag{2}$$

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 logarithm likelihood function is

$$\begin{aligned} l(\rho) = \sum_{i=1}^{s}\{-\ln 2\pi - \frac{1}{2}\ln(1-\rho^2) - \frac{z_{1i}^2 + z_{2i}^2 - 2\rho z_{1i}z_{2i}}{2(1-\rho^2)}\} \\ + \sum_{i=1}^{s}[\ln f(x_1^i) + \ln g(x_2^i)], \end{aligned} \tag{3}$$

where,

$$z_{1i} = \phi^{-1}(F(x_1^i)), z_{2i} = \phi^{-1}(G(x_2^i))\cdot \tag{4}$$

If Moment Estimation is used, then

$$\overset{\wedge}{\rho} = \frac{\frac{1}{s}\sum_{i=1}^{s}x_1^i x_2^i - \bar{X}_1\bar{X}_2}{S_{X_1}^* S_{X_2}^*}, \tag{5}$$

where,

$$\bar{X}_k = \frac{1}{s}\sum_{i=1}^{s}x_k^i, \quad S_{X_k}^* = \sqrt{\frac{1}{s-1}\sum_{i=1}^{s}(x_k^i)^2}, (k = 1, 2). \tag{6}$$

Next, sampling from the distribution function is discussed. Since

$$\begin{aligned} & C(u, v\ ;\rho) \\ & = \phi_\rho(\phi^{-1}(u), \phi^{-1}(v)) \\ & = \int_{-\infty}^{u}\int_{-\infty}^{v}\frac{1}{2\pi\sqrt{1-\rho^2}} \\ & \exp\{-\frac{[\phi^{-1}(s)]^2 + [\phi^{-1}(t)]^2 - 2\rho\phi^{-1}(s)\phi^{-1}(t)}{2(1-\rho^2)}\}dtds \end{aligned} \tag{7}$$

then

$$\begin{aligned} & \omega = C_u(v) \\ & = \partial C(u,v)/\partial u \\ & = e^{-\frac{[\phi^{-1}(u)]^2}{2}}\int_{-\infty}^{v}\frac{1}{2\pi\sqrt{1-\rho^2}}\exp\{-\frac{[\phi^{-1}(t) - \rho\phi^{-1}(u)]^2}{2(1-\rho^2)}\}dt \end{aligned} \tag{8}$$

Therefore,

$$\begin{aligned} & v = \sqrt{1-\rho^2}\phi^{-1}(e^{\frac{[\phi^{-1}(u)]^2}{2}}\omega) + \rho\phi^{-1}(u) \\ & = \sqrt{1-\rho^2}\phi^{-1}(\omega') + \rho\phi^{-1}(u) \end{aligned} \tag{9}$$

So randomly generate vector $(u, \omega') \sim U[0,1]^2$, and then the other random number v was calculated by using (9). The vector (x_1, x_2) can be calculated by using (10).

$$x_1 = \overset{\wedge}{\sigma}_1\phi^{-1}(u) + \overset{\wedge}{\mu}_1、 x_2 = \overset{\wedge}{\sigma}_2\phi^{-1}(v) + \overset{\wedge}{\mu}_2 \tag{10}$$

where,

$$\overset{\wedge}{\mu}_k = \bar{X}_k, \overset{\wedge}{\sigma}_k = S_{X_k}^*, (k = 1, 2). \tag{11}$$

Conclusively, the process for implementing 2-D copula-EDA is as follows:

Step 1: Initialize (***pop***, *N*). Randomly generate initial population ***pop*** with size *N*. set generation count $g \leftarrow 0$.

Step 2: Selection (***pop***, ***spop***, *select-rate*). Select the best *select-rate*×*N* agents form ***pop*** to ***spop*** according to the agents' fitness.

Step 3: copula-generator (***pop***, ***spop***, *mutate-rate*).

S3.1: Construct the distribution model of ***spop***:

1. calculate the sample average and the sample variance for each variable according to (6), 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 (3) or (5), then the copula C is the same as (7);

S3.2: Generate a new population by iterative using proceed generation(*C*, *F*, *G*), where

$$v = C_u^{(-1)}(\omega) = \sqrt{1-\rho^2}\phi^{-1}(\omega) + \rho\phi^{-1}(u)$$

S3.3: randomly generate some agents by the rate *mutate-rate*.

Step 4: Stop if the termination criteria is met.

Step 5: Set $g \leftarrow g + 1$ and go to Step 2.

III. VALIDATION

The following 9 test functions are used to show the behavior of the proposed algorithm. The test functions F_1~F_3 and F_6~F_8 are also used in [11].

$$F_1(x) = -\frac{100}{10^{-5} + \sum_i |y_i|},$$

where $y_1 = x_1, y_i = x_i + y_{i-1} (i \ge 2)$, $x_i \in [-3,3]$, the optimal result is $F_1{}^*(0,0,...0) = -10^7$.

$$F_2(x) = -\frac{100}{10^{-5} + \sum_i |y_i|},$$

Where $y_1 = x_1, y_i = x_i + \sin y_{i-1} (i \ge 2)$, $x_i \in [-3,3]$, the optimal result is $F_2{}^*(0,0,...0) = -10^7$

$$F_3(x) = -\frac{100}{10^{-5} + \sum_i |y_i|},$$

where $y_i = 0.024 \times (i+1) - x_i$, $x_i \in [-3,3]$, the optimal result is $F_3{}^*(0.024 \times 2,...,0.024 \times (n+1)) = -10^7$

$F_4(x) = \sum_i x_i{}^2$,where $x_i \in [-500,500]$, the optimal result is $F_4{}^*(0,0,...0) = 0$

$F_5(x) = 1 + \sum_i (\sin x_i)^2 - 0.1 \exp(-\sum_i x_i{}^2)$, where $x_i \in [-10,10]$, the optimal result is $F_5{}^*(0,0,...0) = 0.9$

$F_6(x) = \sum_i (x_i{}^2 - A\cos(2\pi x_i)) + A)$, where $x_i \in [-5,5]$, the optimal result is $F_6{}^*(0,0,...0) = 0$

$$F_7(x) = \sum_i (418.9829 + x_i \sin\sqrt{|x_i|}),$$

where $x_i \in [-500,500]$, the optimal result is $F_7{}^*(\text{-}420.9687,...,\text{-}420.9687) = 0$.

$$F_8(x) = \sum_i x_i{}^2 - \prod_i \cos(\frac{x_i}{\sqrt{i+1}}),$$

where $x_i \in [-100,100]$, the optimal result is $F_8{}^*(0,0,...0) = -1$.

$$F_9(x) = [1 +$$
$$(x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1{}^2 - 14x_2 + 6x_1x_2 + 3x_2{}^2)]$$
$$\times[30 +$$
$$(2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1{}^2 + 48x_2 - 36x_1x_2 + 27x_2{}^2)]$$
$$\times[30 +$$
$$(2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1{}^2 + 48x_2 - 36x_1x_2 + 27x_2{}^2)],$$

where $x_1, x_2 \in [-2,2]$, the optimal result is $F_9{}^*(0,-1) = 3$

TABLE I

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 II

THE STATICS OF DIFFERENT ALGORITHM AFTER 50 RUNS

	copula_EDA	PBILc	PBILcM
	Mean±Std	Mean±Std	Mean±Std
F_1	-9.7229e+006± 1.5966e+004	-9.9473e+006± 3.1487e+004	-4.4996e+005± 4.4091e+005
F_2	-9.4549e+006± 1.3826e+003	-9.9403e+006± 3.1134e+004	-3.8927e+005± 3.7505e+005
F_3	-9.8119e+006± 1.1698e+003	-9.9519e+006± 2.0524e+004	-9.4466e+004± 6.6105e+004
F_4	3.6721e-012± 2.6923e-012	7.8393e-012± 8.4951e-012	0.2069± 0.8377
F_5	0.9000± 3.0600e-011	0.9000± 3.3826e-014	0.9000± 2.0910e-005
F_6	8.3590e-014± 5.7045e-014	2.2464e-012± 2.6022e-012	5.8463e-004± 6.0017e-004
F_7	0.1596± 0.7787	35.1368± 58.5155	0.3105± 0.2903
F_8	-1.0000± 4.4546e-014	-1.0000± 3.6992e-013	-0.9907± 0.0388
F_9	3.0000± 9.9700e-009	7.8600± 10.4784	3.0116± 0.0206

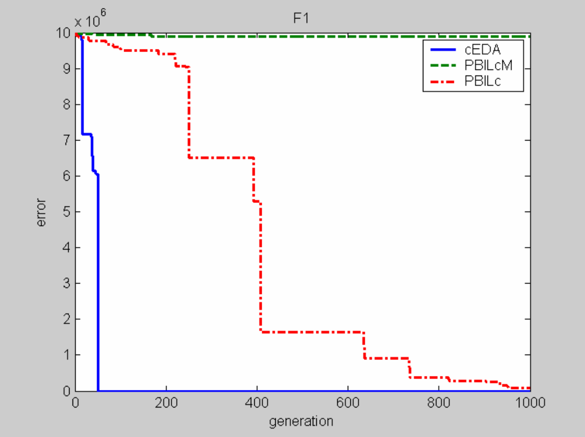

Fig. 2. The performance on test function F_1

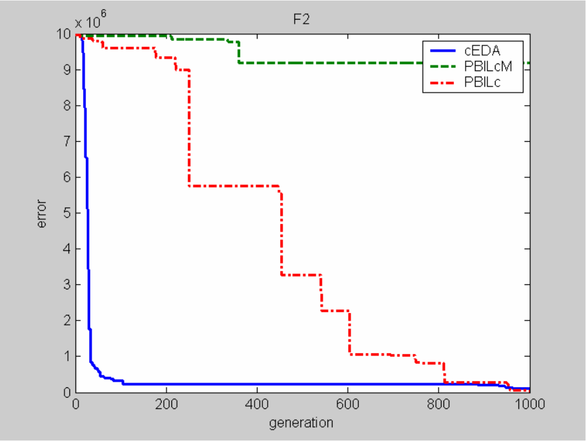

Fig. 3. The performance on test function F_2

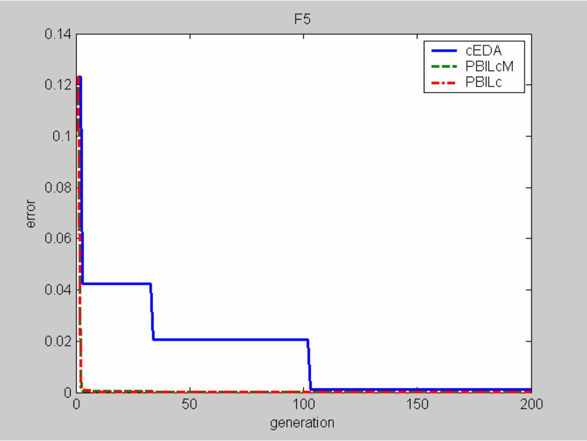

Fig. 6. The performance on test function F_5

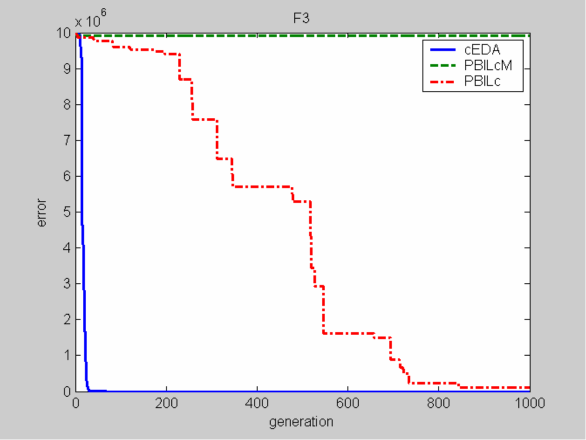

Fig. 4. The performance on test function F_3

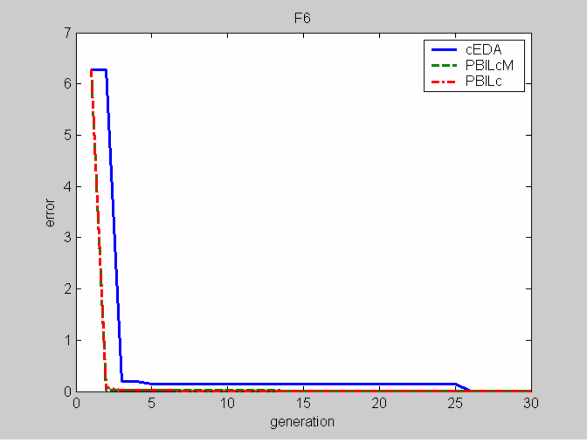

Fig. 7. The performance on test function F_6

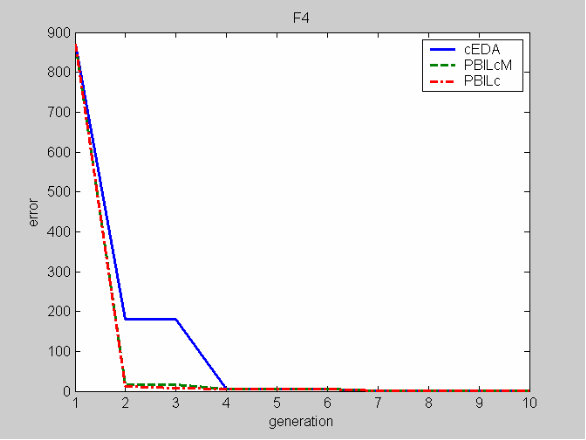

Fig. 5. The performance on test function F_4

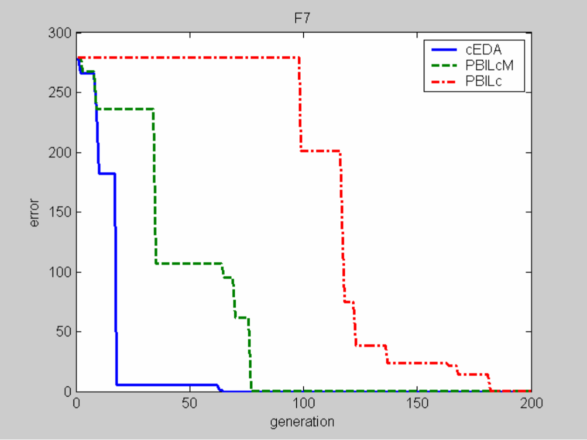

Fig. 8. The performance on test function F_7

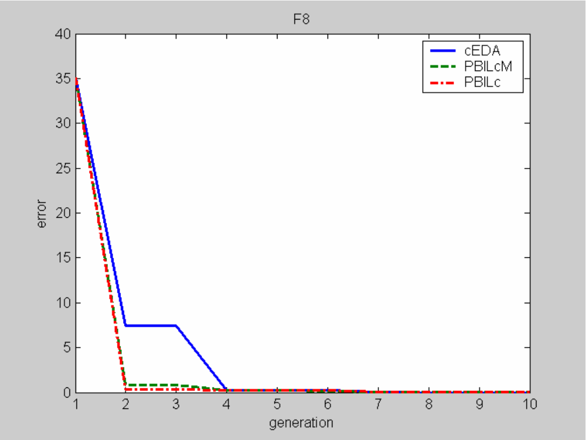

Fig. 9. The performance on test function F_8

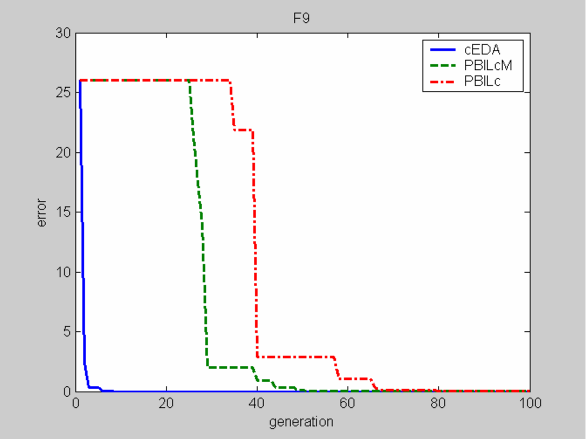

Fig. 10. The performance on test function F_9

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 I and Table II display the experimental results.

copula-EDA is abbreviated to cEDA in Table I and Fig.2 - Fig.10. 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 I. 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 II. 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 begin with the same population in Fig. 2 - Fig. 10. The population in the first generation is same for each algorithm. Error in Fig. 2 - Fig. 10 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 variables are strongly correlative with each other. It is also can be seen from the figure of F_5, F_6 and F_7 that copula-EDA is premature in some cases.

IV. Conclusion

Copula theory provides a good basis for studying multivariate distribution functions. The marginal distributions and the dependence between variables can be researched separately which make the operation easier. Copula-EDA constructs the distribution model based on copula theory. The constructed model reflects the distribution of promising population well, and the computation cost is less than PPCA or BOA. Moreover, different joint distribution functions can be made by selecting different copulas if the marginal distribution functions are given. The experimental results validate this algorithm. The next targets to study is extending this algorithm to high dimensional optimization problems and keeping the algorithm away from prematurity.

References

[1] P. Larranaga, and J. A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Boston: Kluwer Academic Publishers, 2002.

[2] S. Baluja, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," Technical Rep. CMU-CS-94-163, Pittsburgh, PA: Carnegie Mellon University, 1994.

[3] H. Muhlenbein, and G.Paass, "From recombination of genes to the estimation of distributions I. Binary Parameters," in *Proc. PPSN IV*, Berlin, 1996, pp. 178-187.

[4] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," in *Proc. IEEE Conf. Evolutionary Computation*., Indianapolis, USA, 1998. pp. 523-528.

[5] J. S. De Bonet, C. L. Isbell , P. Viola, "MIMIC: Finding optima by estimation probability densities," *Advances in Neural Information Processing Systems* , Cambridge: MIT Press, 1997, pp. 424-430.

[6] W. Zhong, J.Liu, F.Liu, and L. Jiao, "Second order estimation of distribution algorithms based on kalman filter," *Chinese Journal of Computers*, vol. 27, no. 9, pp. 1272-1277, Sept. 2004. (in Chinese)

[7] M. Pelican, and H.Muhlenbein, "The bivariate marginal distribution algorithm," *Advances in Soft Computing-Engineering Design and Manufacturing*, London, Springer-Verlag, 1999, pp. 521-535.

[8] G.Harik, "Linkage Learning via Probabilistic Modeling in the ECGA," Illigal Rep. No.99010, Illinois Genetic Algorithms Lab., University of Illinois, Urbana-Champaign, Illinois, 1999.

[9] H. Muhlenbein, and T.Mahnig, "FDA- a scalable evolutionary algorithm for the optimization of additively decomposed functions," *Evolutionary Computation*, vol. 7, no. 4, pp. 353-376, Winter 1999.

[10] M, Pelikan, D. E. Goldberg, and E.Cantu-Paz, "BOA: the Bayesian optimization algorithm," in Proc. Genetic and Evolutionary Computation Conference (GECCO-99), Orlando, FL, 1999, pp. 525-532.

[11] M. Sebag, and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," in *Proc. PPSN V*, Amsterdam, 1998, pp. 418-427.

[12] P. Larranaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, "Optimization in continuous domains by learning and simulation of Gaussian networks," in *Proc. 2000 Genetic and Evolutionary Computation Conf. Workshop Program*, Las Vegas, Nevada, 2000, pp. 201-204.

[13] D. Y. Cho, and B T. Zhang, “Continuous estimation of distribution algorithms with probabilistic principal component analysis,” in *Proc. IEEE Conf. Evolutionary Computation*, Seoul, Korea, 2001, pp. 521-526.

[14] P. Larranga, R. Etxeberria, J. A. Lozano, and J. M. Pena. “Optimization by Learning and Simulation of Bayesian and Gaussian Networks,” Technical Rep. EHU-KZAA-IK-4/99, University of the Basque Country, Spain,1999.

[15] P. Larranaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, “Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks,” in *Proc. GECCO-2000*, San Francisco, USA, 2001, pp. 201-204.

[16] P. Bosman, D. Thierens. (2008, December 1). IDEAs based on the normal kernels probability density function. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.1984.

[17] M. Gallagher, and M. Frean, “Population-based continuous optimization, probabilistic modelling and mean shift,” *Evolutionary Computation*, vol. 13, no. 1, pp. 29-42, Spring 2005.

[18] J. Ocenasek, S. Kern, N. Hansen, S. Muller, and P. Koumoutsakos. “A Mixed Bayesian Optimization Algorithm with Variance Adaptation,” in *Proc. PPSN VIII*, Birmingham, UK, 2004, pp. 352－361.

[19] Q. Zhang, and H.Muhlenbein, “On the convergence of a class of estimation of distribution algorithms,” *IEEE Trans. Evolutionary Computation*, vol. 8, no.2, pp. 127~136, Apr. 2004.

[20] Q. Zhang. (2008, October 1). On the convergence of a factorized distribution algorithm with truncation selection. [Online]. Available: http://cswww.essex.ac.uk/staff/zhang/EDAWEB/.

[21] R. B.Nelsen, *An Introduction to copula*. Springer-Verlag, New York, 1998.

[22] J. Wu, C. Liu, and X. Qiu, “Dependence measure based on copula theory,” *Journal of Shanxi University(Nat. Sci. Ed.)* , vol. 31, no. 3, pp. 299-302, 2008.（in Chinese）

[23] D. Shi, and Q. Yao, “A method of improving copula fitted to data,” *System Engineering - Theory & Practice*, no.4, pp. 49-54, Apr. 2004. (in Chinese)

[24] Y. Zhang, “copula technique and financial risk analysis,” *Statistical Research*, no.4, pp. 48-51, Apr. 2004. （in Chinese）

[25] M. Bai, L. Sun, “The comparison between three copula-VaR approaches and traditional VaR methods,” *The Journal of Quantitative & Technical Economics*, no. 2 pp. 154-160, Feb. 2007. (in Chinese)

[26] S. Demarta, and A. J. McNeil. (2008, October 1). The t copula and Related copulas [Online]. Available: http://www.math.ethz.ch/~mcneil/ftp/tcopula.pdf.