A modified Alopex-based evolutionary algorithm and its application on parameter estimation

Pengfei He, Postgraduate Student, Key Laboratory of
Advanced Control and Optimization for Chemical
Processes (East China University of Science and
Technology), Ministry of Education, Shanghai, 200237,
E-mail: pfh654116@163.com;

Shaojun Li, Ph.D., professor, Key Laboratory of
Advanced Control and Optimization for Chemical
Processes (East China University of Science and
Technology), Ministry of Education, Shanghai, 200237,
E-mail: lishaojun@ecust.edu.cn;

***Abstract*—In order to improve the efficiency of an Alopex-based evolutionary algorithm (AEA), a modified AEA algorithm (CAEA) which combines copula estimation of distribution algorithm (copula EDA) is introduced in this paper. In view of the inefficiency and the lack of adequate evolutionary information for the population in AEA, a set of competitive and elite solutions are acquired to improve the quality and maintain the diversity of the candidate population by using EDA based on copula. The modified algorithm not only takes advantage of heuristic search of AEA, but also inherits the superiority of rapid convergence of copula EDA. Then 22 benchmark functions are employed to test the performance of CAEA algorithm. Compared with AEA, EDA and differential evolution (DE), the optimization results indicate that the performance of CAEA is significantly superior to that of the other three algorithms, no matter in accuracy or in stability. Furthermore, the modified algorithm is applied to estimating the parameters of a fermentation dynamics model. The results of the comparisons with the other two algorithms illustrate that CAEA algorithm is effective in practical engineering application.**

Keywords—AEA; EDA; Copula; Parameter estimation; Wilcoxon signed test

I. INTRODUCTION

In the past few decades, different kinds of evolutionary algorithms (EAs) had been developed by researchers. Nowadays, evolutionary algorithms are playing important roles in some subjects, especially in the fields of scientific computation and engineering. Amongst them, Genetic algorithm (GA), which was first introduced by Holland, is noteworthy because of its strong robustness and widespread application [1-4]. And it has achieved great advancement in related areas, such as function optimization, network optimization, image processing, and machine learning [5-6]. Based on the principle of natural selection and genetic inheritance, the process of selection and variation (mutation and crossover) form the basis of GA. The former favors the quality, and the latter facilitates diversity and novelty.

Simulated annealing (SA) finds its inspiration in the physical annealing process and utilizes the principles of statistical mechanism, can extricate solutions vectors from local optimum by selecting the “nearby” candidate solution with a probability [7-8]. To some extent, the algorithm of pattern extraction (Alopex) which is introduced in the following paragraph also contains the idea of SA.

Algorithm of pattern extraction (Alopex) [9] was proposed in 1974 and the aim of Alopex was to solve combinatorial optimization and pattern matching problems. The algorithm is inspired by the influence of previous change of independent variables with respect to the objective function, and uses the process parameter ‘temperature’ to control the probability to achieve the search process optimization.

In the past two decades, some pertinent work on Alopex had been done by researchers. Pandya, Sen and Hsu [10] used Alopex to optimize the buffer allocation in ATM switching networks. Alopex algorithm was also used to improve other evolutionary algorithm. Li, Zhang and Qian [11] developed a PSO-Alopex algorithm by embedding Alopex into PSO to train a BP neural network. However, the basic Alopex algorithm is a single point iterative algorithm, which means that only one point can be updated in each iteration and can be less competitive. Based on stochastic idea of Alopex algorithm and swarm intelligence-based evolutionary mechanism, an Alopex based evolutionary algorithm (AEA) was proposed by Shaojun Li and Fei Li [12]. A distinguished feature of AEA is that all variables are updated simultaneously, which is a very contributing factor to improve the efficiency of the algorithm. Meanwhile, the strategy that the generation method of the candidate population simply depends on random initial population is unconvincing and inappropriate. The demerits can result the associated consequences such as low efficiency, low convergence rate and prone to be premature when solving some complex problems.

In order to overcome the defects brought by the previous improper generation method of population, avoid trapping into a local optimum and enhance the performance of AEA, the candidate population is generated and improved by employing estimation of distribution algorithm based on copula theory. What’s more, the newly generated population contains global probabilistic information and maintains its diversity.

978-1-4799-7492-4/15/$31.00 © 2015 IEEE

Eventually, a modified AEA algorithm (CAEA) by copula estimation of distribution algorithm is proposed.

II. Fundamentals of AEA and copula EDA

A. Alopex-based evolutionary algorithm

The alopex-based evolutionary algorithm (AEA) was proposed by Shaojun Li and Fei Li [12]. It is a novel algorithm that not only inherits the primary characteristics of basic EAs, but possesses the merits of gradient methods and simulated annealing algorithm. In the search process, AEA utilizes the change information of the previous variables and function values and contains the idea of gradient descent and Simulated Annealing algorithm to some extent.

Take a minimization of the function as an example. Suppose that there are the *i*-th individuals X_i^k and Y_i^k in two populations P_1 and P_2 , respectively, and they have identical individuals with different arrangement orders at iteration time *k*. In Eq. (1), $F(X_i^k)-F(Y_i^k)$ denotes the difference of the two corresponding function values, where, $X_i^k=(x_{i1}{}^k,x_{i2}{}^k,...,x_{iN}{}^k)$, $Y_i^k=(y_{i1}{}^k,y_{i2}{}^k,...,y_{iN}{}^k)$. C_{ij}^k represents the correlation between the variables and the corresponding function values, where $i=1,...,M$ (M is the number of individual in one population), $j=1,...,N$ (N is the number of variable dimension). The superscript *k* is used to mark the generation of the population.

$$C_{ij}{}^k=(x_{ij}{}^k-y_{ij}{}^k)\cdot[F(X_i^k)-F(Y_i^k)] \tag{1}$$

Annealing temperature T^k influences the algorithm to a large extent and can be calculated by Eq. (2). The annealing strategy adopted here can be calculated and updated once in each iteration.

$$T^k=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}C_{ij}^{\ k} \tag{2}$$

The probability of moving direction p_{ij}^k is given by Eq. (3), and it has influence on the direction of moving step length Δ_{ij} and determines whether the step value will be added or subtracted to the original individuals. Step length Δ_{ij} is presented by Eq. (5).

$$p_{ij}{}^k=\frac{1}{1+e^{(\frac{\pm C_{ij}{}^k}{T^k})}} \tag{3}$$

$$\delta_{ij}{}^k=\begin{cases}1, & p_{ij}{}^k \geq rand(0,1)\\ -1, & otherwise\end{cases} \tag{4}$$

$$\Delta_{ij}{}^k=\left|y_{ij}{}^k-x_{ij}{}^k\right|\cdot\delta_{ij}{}^k\cdot rand(0,1) \tag{5}$$

A newly generated trial variable $(x_{ij}^k)'$ can be calculated according to Eq. (6). Lastly, the update strategy is given by Eq. (7). Compare the newly generated trial solutions with the old ones, the better performing solutions are selected, otherwise, the old ones will be remained.

$$(x_{ij}{}^k)'=x_{ij}{}^k+\Delta_{ij}{}^k \tag{6}$$

$$x_{ij}{}^{k+1}=\begin{cases}(x_{ij}{}^k)', & F((X_i{}^k)')<F(X_i{}^k)\\ x_{ij}{}^k, & otherwise\end{cases} \tag{7}$$

In each iteration, two populations are generated so as to execute the related operations from Eq. (1) to Eq. (7). More specifically, at the initial stage of search, high temperature value helps to extend searching scope, and the possibility of finding the global optimum can be increased. With temperature decreasing gradually, the stochastic search turns into a deterministic one. During the process, each variable of population not only moves to a better direction, but has opportunities to the opposite one. In this circumstance, the algorithm performs better in climbing the hill.

B. Copula EDA

Since it was proposed in 1996, estimation of distribution algorithms (EDAs) have become hot research topics in the field of evolutionary computing [13-15]. EDAs are stochastic optimizations with the main characteristics of efficiently exploring the space of candidate solutions by building and sampling an explicit probabilistic model constructed from promising solutions found so far, instead of using conventional evolutionary operators such as crossover and mutation. The use of probabilistic learning model is to accurately identify the features of promising candidate solutions from the population, so as to guide the search toward promising regions of the search space.

In this paper, the method of selection is truncation selection. For the issue of building a probability distribution model, the joint density function of N-dimensional variables here is factorized as a product of N parameters with independent normal distributions. The joint density function can be calculated by Eq. (8):

$$f(x)=\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi}\sigma_i}\exp[-\frac{1}{2}(\frac{x_i-\mu_i}{\sigma_i})^2] \tag{8}$$

mean of the *i*-th individual μ_i and standard deviation σ_i can be estimated as follows [16]:

$$\hat{\mu}_i=\overline{X}_i=\frac{1}{NS}\sum_{t=1}^{NS}x_{t,i}\ , \qquad \hat{\sigma}_i=\sqrt{\frac{1}{NS}\sum_{t=1}^{NS}(x_{t,i}-\overline{X}_i)^2} \tag{9}$$

where, NS is the number of selected solutions.

Inheriting the basic characteristics of EDA, and based on the copula theory, copula estimation of distribution algorithm (copula EDA) is introduced in this paper.

The copula theory is Sklar's theorem which can be described as follows[17-18]: Suppose that there exists an n-dimensional joint cumulative distribution function F with marginal distribution functions $F_1F_2,..F_n$. Then there is an n-dimensional copula function C for all x ∈ R^n,

$$F(x_1,x_2,...,x_n)=C(F_1(x_1),F_2(x_2),...,F_n(x_n)) \tag{10}$$

If the random variables' marginal distribution functions $F_1F_2,\dots F_n$ are all continuous, then C is unique. Conversely, if C is an n-copula, and $F_1F_2,\dots F_n$ are distribution functions, then the function F is an n-dimensional distribution function with margins $F_1F_2,\dots F_n$.

The Sklar's theorem indicates that the joint distribution function F can be constructed by one dimensional marginal distribution of each random variable and a corresponding copula function. On the ground of the theorem, the estimation of joint distribution can be transformed into the estimation of marginal distribution of each random variable and the sampling of copula function.

Suppose that the joint distribution of n-dimensional random variables is F, and copula function is C, one dimensional marginal distribution F_i, $(i = 1,2,\dots n)$. In order to acquire individuals $x_1x_2,\dots x_n$ that obey F, the first work is to estimate the marginal distribution for each random variable. The second is to sample a vector $u_1u_2,\dots u_n$ from C according to the Sklar's theorem. Subsequently, the samples $x_1x_2,\dots x_n$ which obey the joint distribution F can be calculated by $x_i = F_i^{-1}(u_i)$ in accordance with the transformation form of Sklar's theorem $C(u_1u_2,\dots u_n) = F(F_1^{-1}(u_1)F_2^{-1}(u_2),\dots F_n^{-1}(u_n))$ [18-20], where, $u_i \in [0,1]$, and $F_i^{-1}$is the inverse function of each marginal distribution. Archimedean copulas can also be called exchangeable copulas when $C(u)= \varphi^{-1}(\varphi(u_1) + \varphi(u_2) + \cdots + \varphi(u_n))$, φ denotes the generator of copula function which is monotonically decreasing [20].

The main idea of copula theory is to decompose the multivariate joint distribution into each univariate marginal distribution and a copula function, instead of estimating the joint distribution directly. Thus, copula theory used in EDA can simplify the operations of the latter algorithm. In the process of EDA based on copula, marginal distribution functions need to be estimated, and the new individuals which obey the joint distribution are obtained by sampling from copula and then calculating the inverse of the univariate marginal distribution function.

Based on the relationship between the generator of copula and distribution function, an algorithm samples from copula was proposed by Marshall and Olkin can be described as follows [21-22].

(1) Sample $v \sim F = LS^{-1}[\varphi^{-1}]$, where $LS^{-1}[\varphi^{-1}]$is the inverse Laplace transform of the generator φ^{-1}.
(2) Sample $y_i \sim U(0,1)$, $i = 1,2,\dots n$.
(3) Calculate $u_i = \varphi^{-1}(\ln(y_i)/\upsilon)$, then vector $(u_1,u_2,\dots,u_n)$ could be gained.

This method is simple and practicable and it is appropriate for Archimedean copulas including Clayton copula, Gumbel copula and Frank copula [22-23]. In this paper, Clayton copula is selected, the generating function $\varphi^{-1} = (1+t)^{-\psi}$, the inverse Laplace transformation of the generator is $F(v) = e^{-v} \cdot v^{\psi-1} / \Gamma(\psi)$, where, $\psi = 1/\theta$.

III. Fusion of AEA and copula EDA algorithm

Though the AEA algorithm performs well in some optimization domains, it is evident that the algorithm exposes some problems such as falling into local minima, converging too early and so on. Further research has been done to produce candidates more reasonably based on the mechanism of AEA.

In AEA, two different individuals are chosen randomly, Alopex operation is executed and the newly individual is generated, substitute or reserve the old individual depends on which performs better. In this paper, more excellent individuals can be acquired as an elitist set by a novel idea that an estimation of distribution algorithm based on copula is embedded into AEA. In the newly algorithm (CAEA), EDA based on copula theory is used to generate more prominent candidate population, which is then passed to AEA for related calculation, with more global search information provided. In the point of evolutionary strategy, both of the two algorithms have an impact on the evolutionary direction of population and the characters of CAEA. CAEA not only takes advantage of rapid convergence of copula EDA, and its unique simplified handling model for complex construction of variables on the macro level, but has the merits of strong local searching ability and high searching precision of AEA on the micro level.

In this paper, the minimization of function is focused on, and the detailed implementation procedures of CAEA can be described as follows:

Step1: Initialize the population randomly within the search scope and parameters such as population size, variable dimension, maximum number of iteration, truncation selection probability and so on. Calculate the value of objective function. Set $k = 1$.

Step2: Sort the function value in ascending order, select promising solution set s by truncation selection. Estimate Gaussian marginal distribution of each variable based on the promising solution set $\hat{s}$: Calculate average value μ_i and standard deviation σ_i of each variable in s according to Eq. (9).

Step3: Generate new solutions: Firstly, sample $v \sim F = LS^{-1}[\varphi^{-1}]$; Secondly, sample $y_i \sim U(0,1)$, and the vector $(u_1u_2,\dots u_n)$ is calculated by $u_i = \varphi^{-1}(\ln(y_i)/\upsilon)$; Lastly, the newly individuals are generated by calculating $x_i = F_i^{-1}(u_i;\hat{\mu}_i,\hat{\sigma}_i)$.

Step4: Make judgments on whether it is beyond the feasible region or not for individuals in the set, make some corrections if necessary. Update the old population by the replacement of the new individuals with the rate of r and the newest population P_1 is acquired.

Step5: Population P_2 is generated as follows: Suppose that the individuals in population P_1 have a sequence number from 1 to u, choose an integer w within $[2,u]$, then rearrange the individual order in population P_1

according to the sequence [*w*,*w*+1,…,*u*,1,2,…,*w*-1] to form the population P_2 .

Step6: In population P_1 and P_2 , calculate the difference of the two objective function values. And $C_{ij}^{\ k}$, annealing temperature T^k and the probability $p_{ij}^{\ k}$ is available according to Eq.(1)~Eq.(3).

Step7: Figure out the walking step $\Delta_{ij}^{\ k}$, generate a trial variable $(x_{\mathrm{ij}}^{\mathrm{k}})'$ in P_1 successively, from Eq. (4) to Eq. (6). The changed population P_1 is renamed as P_3 .

Step8: Update population P_1 with P_3 . If the new trial solution $(X_i^{\ k})'$ in P_3 composed by all the newly generated trial variables, performs better than the corresponding individual $X_i^{\ k}$ in P_1 , then the old individual will be replaced by the new corresponding one according to Eq. (7).

Step9: Output the result if the termination criterion meets. Otherwise, $k = k + 1$ and go to step 2.

By embedding copula EDA into AEA, more evolutionary information is provided and the diversity of the population is enhanced. Thus, the probability of finding the optimal solution can be increased. Therefore, both heuristic information which is similar to gradient descent comes from AEA, the advantage of fast convergence and evolutionary information from copula EDA, are integrated to guide the population towards a more promising search domain.

IV. Computational results

A. Benchmark functions

In this section, 22 real-valued standard benchmark functions [24] are used in this paper. Some of them are multimodal and highly nonlinear and have lots of local optimum, such as f1, f4 and f8. Complex unimodal functions including Rosenbrock function (f9), whose global optimum is inside a long, narrow, parabolic shaped flat valley, are also included. In some test functions, such as f5 and f16, variables are related and extremely complicated. All these functions have their own distinct characteristics which can help to evaluate algorithms' ability comprehensively from different aspects.

B. Experimental setup

In the experiments of performance comparison between CAEA and three other algorithms (EDA, DE and AEA), the population size and dimension are set at 100 and 10, respectively. According to the literature [24], it can be considered to be a successful run if the value obtained by the algorithm is within 1% accuracy of the known optimal solution. The maximum number of iteration is 2000. Truncation probability is 0.5 and replacement rate is 0.9. Moreover, the maximum of 2000 generations or no improvement observed in the best individual in consecutive 100 generations is regarded as the termination condition.

TABLE I. Comparison of the optimization result of four algorithms

F	DEA			DE			AEA			CAEA		
	RS	*ABFV*	*DS*	*RS*	*ABFV*	*DS*	*RS*	*ABFV*	*DS*	*RS*	*ABFV*	*DS*
f_1	84	8.88E-16	0	1	4.44E-15	0	54	4.35E-15	6.56E-16	87	2.32E-15	6.81E-17
f_2	100	0	0	100	0	0	100	0	0	100	0	0
f_3	87	0	0	100	0	0	100	2.13E-18	1.68E-17	100	0	0
f_4	92	9.03E-03	2.86E-03	0	-	-	75	7.57E-03	1.41E-03	99	6.49E-05	0
f_5	96	4.71E-32	4.71E-47	100	4.71E-32	4.95E-47	100	4.71E-32	4.95E-47	100	4.71E-32	4.95E-47
f_6	100	1.35E-32	5.56E-47	100	1.35E-32	2.48E-47	100	1.35E-32	2.48E-47	100	1.35E-32	2.48E-47
f_7	95	-4.70E-04	0	100	-4.70E-04	6.60E-15	100	-4.70E-04	2.44E-15	100	-4.70E-04	2.38E-15
f_8	100	6.63E-02	0.25	6	0	0	21	0	0	89	0	0
f_9	75	9.59	7.42E-01	98	1.15E-07	7.94E-08	100	4.37E-01	8.23E-01	100	3.85E-01	4.89E-01
f_{10}	0	-	-	100	1.32E-04	1.98E-13	100	1.27E-04	1.96E-13	100	1.27E-04	9.09E-14
f_{11}	100	0	0	100	0	0	100	0	0	100	0	0
f_{12}	89	3.86E-34	5.76E-34	41	6.89E-08	8.47E-08	100	7.45E-44	6.38E-44	100	1.14E-51	8.81E-52
f_{13}	100	4.63E-56	9.12E-56	100	2.53E-45	6.90E-46	100	6.89E-101	3.54E-101	100	0	0
f_{14}	100	1.34E-55	2.97E-55	100	1.39E-44	2.66E-44	100	5.63E-101	2.18E-101	100	7.14E-127	0
f_{15}	78	2.90E-27	3.94E-27	100	3.95E-24	1.89E-24	100	6.81E-71	3.98E-70	100	2.57E-101	4.64E-101
f_{16}	0	-	-	100	-1.21E-12	6.03E-13	100	1.23E-08	1.72E-08	100	4.37E-13	1.09E-12
f_{17}	0	-	-	0	-	-	0	-	-	0	-	-
f_{18}	100	0	0	100	0	0	100	0	0	100	0	0
f_{19}	97	9.72E-03	8.85E-06	23	9.78E-03	6.70E-05	100	9.72E-03	3.43E-08	100	9.72E-03	1.81E-09
f_{20}	90	2.12E-56	6.92E-56	100	1.61E-46	2.56E-46	100	7.14E-101	2.32E-101	100	4.60E-126	2.30E-126
f_{21}	100	2.28E-56	4.28E-55	100	4.81E-45	4.32E-45	100	6.69E-101	3.87E-101	100	4.94E-125	0
f_{22}	100	4.87E-51	1.14E-51	100	8.34E-41	8.10E-41	100	4.12E-53	7.15E-53	100	5.47E-124	2.39E-123
NNWR	9	8	7	15	9	9	18	9	6	21	20	17

The program of DE algorithm is adopted in this paper (at: http://www1.icsi.berkeley.edu/~storn/code.html#matl). The parameters in devec3 are set as follows: Crossover probability, CR is equal to 0.5; step size, F is 0.8; DE algorithm strategy, DE/rand/1.

All the algorithms are implemented in MATLAB code and carried on a DELL computer with Intel Core i7-2600 CPU with frequency 3.4 GHz, 8GB RAM under windows 7, 32-bit operation system.

C. *Comparison and Analysis*

In this group of experiments, 3 kinds of measurement indexes are listed: rate of success (RS), average best function values (ABFV) and standard deviations (SD) of 100 independent tests of algorithms. RS observes the reliability, ABFV evaluates the quality of solutions and SD measures the stability of algorithms. The unsuccessful tests are marked with "-". For the purpose of a holistic comparison, the number of no worse results (NNWR) with other algorithms is also listed at the bottom of Table 1.

Table 1 shows the test results gained by EDA, DE, AEA and CAEA. Compared with other three algorithms, the rate of successful convergence of CAEA reaches 100% for the majority of the functions in 100 independent runs, gaining better results than EDA and DE on the majority of test functions. As can be seen from Table 1, compared with EDA and DE, the performance of CAEA algorithm is dramatically improved, and CAEA gives smaller values in most cases, no matter in terms of ABFV or DS. As for the comparison with AEA, CAEA performs better while focusing on RS, especially on f1 (87 given by CAEA and 54 given by AEA) and f8 (89 by CAEA and 21 by AEA), it is easy to get stuck in local optimum. Concerning on ABFV and SD, the results have been improved to varying degrees on different functions, especially on f3, f13, f14, f15, f21 and f22, the improvement of the solutions is prominent and obvious. At last but not least, The NNWR of all the indicators given by CAEA are better than the other three algorithms'. The number of no worse ABFV obtained by CAEA is 20, while that by AEA, DE and PSO are only 9/9/8, respectively. The number of no worse SD gained by CAEA is 17, while those obtained by the other three algorithms are 6/9/7, respectively. In briefly, the test results demonstrate not only the validity of CAEA clearly, but the superiority of the algorithm over AEA, DE and EDA. Moreover, by contrasting EDA, AEA and CAEA data, the conclusion can be made: by fusing copula EDA into AEA, both of the two algorithms' performance is improved, and the operations are effective and convictive.

V. FERMENTATION DYNAMICS PARAMETER ESTIMATION USING CAEA

Fermentation dynamics is an important part of biochemistry reaction engineering and also a vital basis for the optimization of fermentation process [25]. The aim of fermentation dynamics is to analyze the interaction and the changes with time between environmental factors and metabolism activity of microorganism. It is of great significance and of practical value for the establishment of accurate model of fermentation dynamics, the controlling and optimizing of the reaction process. In order to reveal the actual reaction process, the first priority of establishing a reliable model is parameter estimation.

Fermentation dynamics is composed of three parts: the dynamics model of microbial growth, the dynamics model of substrate consumption and the dynamics model of product. The model based on Logistic model, proposed by Luedeking-Piret which is simple and widely used in this area, which can be elaborated from Eq. (13)-(15). Here, μ_m, X_max, α, β, k_1, k_2 are 6 parameters to be optimized. *X* is the mass concentration of bacteria; *P* is the mass concentration of product (g/L); *S* denotes the mass concentration of substrate (g/L); μ_m is the biggest growth rate of bacteria; X_max is the largest mass concentration of bacteria (g/L); α and β denote the coefficient about growth and the coefficient about bacteria (h^{-1}), respectively; k_1 is bacteria yield coefficient relative to substrate and k_2 is the product yield coefficient relative to substrate.

$$\mathrm{X} = \frac{\mathrm{X}_0 X_{max} \exp(\mu_m t)}{X_{max} - \mathrm{X}_0 + \mathrm{X}_0 \exp(\mu_m t)} \tag{13}$$

$$\mathrm{P} = \mathrm{P}_0 + \alpha(\mathrm{X} - \mathrm{X}_0) + \frac{\beta X_{max}}{\mu_m} \ln(\frac{X_{max} \underline{=} \mathrm{X}_0 + \mathrm{X}_0 \exp(\mu_m t)}{X_{max}}) \tag{14}$$

$$\mathrm{S} = \mathrm{S}_0 + (\mathrm{X} - \mathrm{X}_0)(k_1 + \alpha k_2) \\ - \frac{k_2 \beta X_{max}}{\mu_m} \ln\left(\frac{X_{max} - X_0 + X_0 exp(\mu_m t)}{X_{max}}\right) \tag{15}$$

The goal of the optimization is to minimize the square of the difference between experimental values and estimated values of *X*, *P* and *S*. The objective function to be optimized is the sum of the square errors between the measured values and the estimated values as given by Eq.(16). Where, *X*, *P* and *S* are the estimated values and *n* is the number of the observed data. The experimental data used here are cited from [25]. Here, an improved Gauss-Newton method [25] and multiple genetic algorithms [26] are used for comparison.

$$\min SDS = \sum_{i=1}^{n} (\left(X(i) - \hat{X}(i)\right)^2 + \left(P(i) - \hat{P}(i)\right)^2 + \left(S(i) - \hat{S}(i)\right)^2) \tag{16}$$

In this section, the proposed CAEA algorithm is used to estimate the 6 kinetics parameters. The concrete results are provided in Table 2. Eventually, the value of the objective function calculated by CAEA is 40.23, which is much smaller than results obtained by the other two algorithms. Above all, the conclusion can be drawn that CAEA has a better application in the parameter estimation of fermentation dynamics model.

TABLE II. PARAMETER ESTIMATION RESULTS OF FERMENTATION DYNAMICS MODELS GIVEN BY THREE ALGORITHMS

Method	μ_m	$X_{\max}$	α	β	k_1	k_2	SDS
GA [26]	0.141	22.66	0.0840	0.0050	0.0030	17.347	210.83
Gauss-Newton[25]	0.147	22.60	0.0052	0.0888	0.0052	17.94	104.32
CAEA	0.1418	22.5332	0.07478	0.00556	6.81E-05	17.35	40.23

VI. CONCLUSIONS

In this paper, the candidate population is generated by using of copula EDA. Thus, the population in CAEA contains global search information and maintains the diversity. The test results of 22 benchmark functions reveal that the CAEA is feasible and effective. At last, CAEA is applied to a practical application, and the promising results are acquired.

More improvements need to be done for the algorithm, in spite of the excellent results. One of our future works will focus on further improvement on more reasonable mode of the annealing temperature. How to solve the problem such as Neumaier function is another task. Based on CAEA, solving constraint optimization problems is another work in the future.

ACKNOWLEDGMENTS

The authors of this paper appreciate the National Natural Science Foundation of China (under Project No.21176072) and the Fundamental Research Funds for the Central Universities for their financial support.

REFERENCES

[1] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press, vol. 183.

[2] Jawahar, N., & Balaji, N. (2012). A genetic algorithm based heuristic to the multiperiod fixed charge distribution problem. Applied Soft Computing, 12(2), 682–699.

[3] Hajiaghaei-Keshteli, M., Molla-Alizadeh-Zavardehi, S., & TavakkoliMoghaddam, R. (2010). Addressing a nonlinear fixed charge transportation problem using a spanning tree based genetic algorithm. Computers & Industrial Engineering, 59(2), 259–271.

[4] Ojha, A., Das, B., Mondal, S., & Maiti, M. (2010). A solid transportation problem for an item with fixed charge, vehicle cost and price discounted varying charge using genetic algorithm. Applied Soft Computing, 10, 100–110.

[5] Peng, J., Meng, F., & Ai, Y. (2013). Time-dependent fermentation control strategies for enhancing synthesis of marine bacteriocin 1701 using artificial neural network and genetic algorithm. Bioresource Technology, 138, 345–352.

[6] Antony Arokia Durai Raj, K., & Rajendran, C. (2012). A genetic algorithm for solving the fixed-charge transportation model: two-stage problem. Computers & Operations Research, 39(9), 2016–2032.

[7] Balaji, N., & Jawahar, N. (2010). A simulated annealing algorithm for a two-stage fixed charge distribution problem of a supply chain. International Journal of Operational Research, 7(2), 192–215.

[8] Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. European Journal of Operational Research, 149(2), 268–281.

[9] Harth, E., & Tzanakou, E. (1974). Alopex: A stochastic method for determining visual receptive fields. Vision Research, 14(12), 1475-1482.

[10] Pandya, A. S., Sen, E., & Hsu, S. (1999). Buffer allocation optimization in ATM switching networks using ALOPEX algorithm. Neuro computing, 24 (1-3), 1-11.

[11] Li, S.J., Zhang, X.J., & Qian, F. (2005). Soft sensing modeling via artificial neural network based on PSO-Alopex. Machine Learning and Cynetics. Proceedings of 2005 International Conference on. (pp. 4210-4215).

[12] Li, S.J., & Li, F. (2011). Alopex- based Evolutionary Optimization Algorithms and Its Application to Reaction Kinetic Parameter Estimation. Computers& Industrial Engineering, 60(2), 341-348.

[13] Christopher Expósito-Izquierdo, José Luis González-Velarde, Belén Melián-Batista, & Marcos Moreno-Vega, J. (2013). Hybrid Estimation of Distribution Algorithm for the Quay Crane Scheduling Problem. Applied Soft Computing, 13(10), 4063-4076.

[14] Larrañaga, P., & Lozano, J.A. (2002). Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Boston: Kluwer Academic Publisher.

[15] Mühlenbein, H., & Paaß, G. (1996). From Recombination of Genes to the Estimation of Distributions. Lecture Notes in Computer Science, Parallel problems Solving from Nature, 1411. Heidelberg: Springer. (pp. 178–187).

[16] Sun, J., Zhang, Q., & Tsang, EPK. (2005). DE/EDA: A new evolutionary algorithm for global optimization. Information Science, 169(3-4), 249-262.

[17] Liebscher, E. (2008). Construction of asymmetric multivariate copulas. Journal of Multivariate Analysis, 99(10), 2234-2250.

[18] Nelsen, R. B. (2006). An Introduction to Copulas. (2nd ed.). New York: Springer.

[19] Wu, F., Valdez, E. A., & Sherris, M. (2006). Simulating Exchangeable Multivariate Archimedean Copulas and its Application. Journal of Statistical Computation and Simulation, 78(6), 567-581.

[20] Whelan, N. (2004). Sampling from Archimedean copulas. Quantitative Finance, 4(3), 339-352.

[21] Marshall, A. W., & Olkin, I. (1988). Families of Multivariate Distributions. Journal of the American Statistical Association, 83(403), 834-841.

[22] Hofert, M. (2008). Sampling Archimedean copulas. Computational Statistics & Data Analysis, 52(12), 5136-5174.

[23] Wang, L. F., Zeng, J. C., & Hong, Y. (2009). Estimation of Distribution Algorithm Based on Archimedean Copulas. Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GECS 2009). ACM, New York, (pp. 993-996).

[24] Deep, K., & Das, K. N. (2008). Quadratic approximation based hybrid genetic algorithm. Applied Mathematics and Computation, 203(1), 86–98.

[25] Liu, D.L., Xie, S.J., & Guan, X.Y. (2010). Parameter estimation of fermentation dynamics models by adaptive domain with multiple genetic algorithms. Computing Technology and Automation, 29(3), 18-23.

[26] Song, W.J., Chen, N., & Wang, J. (2002). The kinetics of batch fermentation of L-tryptophan-production strain. Journal of Wuxi University of Light Industry, 21(4), 340-343.