2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

A Novel Hybrid Differential Evolution-Estimation of Distribution Algorithm for Dynamic Optimization Problem

Xiangman Song
The Logistics Institute
Northeastern University
Shenyang, China
xmsong@tli.neu.edu.cn

Lixin Tang
The Logistics Institute
Northeastern University
Shenyang, China
lxtang@tli.neu.edu.cn

Abstract—**In many engineering applications, the dynamic optimization problems with Ordinary Differential Equations (ODE) or Differential Algebraic Equations (DAE) constraints are encountered frequently. These types of problems are solved difficultly because of the characteristic of their nonlinear, multidimensional and multimodal. In this paper, a novel hybrid Differential Evolution (DE) and Estimation of Distribution Algorithm (EDA) is proposed for the dynamic optimization problems. A novel hybrid scheme based on DE and EDA (DE-EDA) is designed to generate the offspring population. Using the DE-EDA, the population can reach a promising area in which the optimal solution is located speedily. A modified mutation scheme is proposed which can increase the diversity of the population. In addition, the modeling and sampling scheme based on empirical Copula is used to improve the speed of modeling and sampling. Eight optimal control optimization problems and one parameter estimation problem are tested to measure the performance of the algorithm. Experimental results show that the algorithm is feasible and effective.**

Keywords—dynamic optimization; estimation of distribution algorithm; differential evolution; empirical Copula

I. INTRODUCTION

Dynamic optimization plays shrugs role in the complex process of industrial production. It is used to solve many off-line and on-line engineering tasks, including the configuration of a batch operation, parameter estimation, model development and identification of dynamic systems, optimal design and control of systems and the optimization problem of the solution in the nonlinear model predictive control. The dynamic optimization problem is very common especially in the field of chemical engineering applications.

Usually it is very difficult to solve these types of problems because they are always highly nonlinear and multidimensional, also due to the existence of the constraints on both state and control variables at the same time [1-3]. In the past several years, different methods have been developed to solve the dynamic optimization problems. These algorithms can be divided into four categories, including the dynamic programming or the gradient algorithms based on the Hamiltonian function [4], the indirect or the calculus of variations approach based on Pontryagin's Maximum Principle, the direct methods [5] and the evolutionary algorithms based on the population. Most of the traditional algorithms based on gradient information relying on the degree of nonlinearity of the problem and the initial guesses for state and adjoint variables, have the possibility of getting trapped at the local optimum [6].

In the direct methods, the dynamic optimization problems can be solved applying non-linear programming (NLP) solvers. These methods can be divided into two classes, the sequential and simultaneous strategies. The advantage of the former strategy lies in its reduced dimensionality of the optimization problem. However, it cannot solve the problems that contain state variable constraints directly. Moreover, it is difficult to treat the unstable systems [5]. On the other hand, in the simultaneous strategy, both the state and control profiles are discretized. The typical discretization methods have direct multiple shooting and collocation on finite differences [7-9]. Though simultaneous strategy is easy to parallelize and robust handling of path and terminal constraints, as the number of variables and complexity of the problems increase, the method become complicated [4].

Evolutionary algorithms (EAs) have become very popular for dynamic optimization problems recently. Genetic algorithms (GA), Differential evolution (DE) and ant colony algorithm have been applied for dynamic optimization problems of some chemical systems [4,10-14,].With the extensive research of intelligent algorithm, a trigonometric differential evolution approach was proposed for the optimization of dynamic systems [3].

Applying EAs to solve the dynamic optimization problems, most of researchers used a real-coded genome to represent a feasible control profile that was divided into several stages through discretizing time interval. In their research, the interrelation between the control variables in each stage was not considered.

If the interrelation is considered, the rate of convergence will increase. In [15], estimation of distribution algorithm

(EDA) was presented. It is evolutionary algorithm based on estimation and sampling from probabilistic models, and able to make use of the interrelation between the variables, which traditional EAs can't do. Several improved EDAs have been proposed recent years. In [16], a combination algorithm of DE and EDA was proposed for the global continuous optimization problem, but the algorithm didn't use the crossover scheme.

This paper proposes a novel hybrid algorithm combining DE and EDA (DE-EDA) for solving dynamic optimization problems encountered in chemical engineering. In this paper, eight optimal control problems and one parameter estimation problem taken from recent literature will be solved.

The rest of this paper is organized as follows. Section II presents a formulation of the dynamic optimization problem. Section III describes the fundamental ideas of the DE-EDA algorithm. Experimental results are shown in Section IV. Finally, conclusions are given in section V.

II. THE DYNAMIC OPTIMIZATION PROBLEM

A. *The Mathematical Description of the Dynamic Optimization Problem*

In general terms, the dynamic optimization problem can be stated in the following form:

$$\min_{x(t),u(t),z(t),p,t} J(t_f) = \varphi(x(t_f), p, t_f) + \int_{t_0}^{t_f} F(x(t),u(t),z(t),p,t)dt \quad (1)$$

subject to a set of differential and algebraic equations (DAEs)

$$\dot{x}(t) = f(x(t),u(t),z(t),p,t) \quad (2)$$

$$h(x(t),u(t),z(t),p,t) = 0 \quad (3)$$

with the initial conditions

$$x(t_0) = x_0 \quad (4)$$

Here, x, z and u are related to the time parameter of t, which satisfies the equation (5). In addition, x and z represent the differential and the algebraic state variables respectively, u is the control variables, whereas p is the vector of time independent parameters.

$$t \in [t_0, t_f] \quad (5)$$

Furthermore, the equation (6) has to be satisfied.

$$r(x(t_f)) = 0 \quad (6)$$

In addition, the state and control inequality constraints are presented as follows, which the x, z and u must be satisfies.

$$g(x(t),u(t),z(t),p,t) \le 0 \quad (7)$$

$$x(t)^L \le x(t) \le x(t)^U \quad (8)$$

$$u(t)^L \le u(t) \le u(t)^U \quad (9)$$

B. *Control Profile Discretization*

In the dynamic optimization, the time interval $[t_0, t_f]$ is divided into D stages, and in the ith time stage $[t_{i-1}, t_i]$, $i = 1,\ldots,D$, $t_D = t_f$, the control profile is approximated with a one-dimensional variable x_i. So for the whole time interval $[t_1, t_f]$, the control profile is presented a D-dimension vector $x = (x_1, x_2, \ldots, x_D)$, $i = 1,2,\ldots,D$. The vector will be served as an individual of the population in our proposed algorithm. In fact, the number of the stages affects the performance of our algorithm. So the D is critical control parameters in our algorithm.

III. OUR PROPOSED ALGORITHM FOR DYNAMIC OPTIMIZATION PROBLEM

In this section, a novel hybrid algorithm based on DE and EDA is presented for dynamic optimization. In proposed algorithm, the population obtained by DE is used as the initial population of the EDA for learning. In addition, the mutation procedure of the basic DE is modified. In proposed algorithm, Copula-based multivariate EDA is employed, where the margins distribution is the empirical distribution. In the following, after description of the basic DE and EDA, a novel hybrid DE-EDA algorithm will be presented for dynamic optimization.

A. *A Brief Introduction to Differential Evolution*

There are several versions of DE algorithm [17]. The algorithm we proposed in this paper is based on the version of DE/best/2/bin. The particular scheme is presented as follows:

DE maintains a population of NP candidate solutions in every generation, where each solution can be represented as a D-dimensional control vector. In generation G, we denote the ith individual of the population as $X_{i,G}, i = 1,2,3..., NP$. The population is evolved and improved generation by generation. In each generation, a new population is generated through the mutation operator and crossover operator, and the individual of the offspring is called trial vector. A selection scheme is used to decide which one to get into the next generation, parent or offspring. The evolution process moves on until a termination criterion is met (e.g., the best objective function value of the current population is better than the given value, or the number of iterations is equal to the given the maximum value). The details are presented as follows.

1) Mutation: In the version of DE/best/2/bin, the mutation operation can be described that two weighted differential vectors obtained from the current population are added to the third vector which is the best individual having the best value. At generation G, the mutation individual $V_{i,G}$ is generated relied on four randomly selected individuals and the best individual $X_{Best,G}$ as follows:

$$\begin{aligned} V_{i,G} = X_{Best,G} + F*(X_{r1,G} - X_{r2,G}) \\ + F*(X_{r3,G} - X_{r4,G}) \end{aligned} \tag{10}$$

Where $i = 1,2,3...,NP$, the randomly selected indices $r1$, $r2$, $r3$ and $r4$ belong to the integer set $\{1,2,\ldots,NP\}$ and satisfy the expression $r1 \neq r2 \neq r3 \neq r4 \neq i$. In addition, F is the scale factor and $F \in [0,1]$.

2) Crossover: After the mutation operation, the trial individual $U_{i,G+1} = (u_{1,i,G+1},\ldots,u_{D,i,G+1})$ is generated from the target individual $X_{i,G+1} = (x_{1,i,G+1},\ldots,x_{D,i,G+1})$ and the mutation individual $V_{i,G+1} = (v_{1,i,G+1},\ldots,v_{D,i,G+1})$ through the crossover operation described as follows:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & if\ rand_j \le CR\ \ or\ \ j = k \\ x_{j,i,G+1} & otherwise \end{cases} \tag{11}$$

Where $j = 1,\ldots,D$, k is a random vector's index and $k \in \{1,\ldots,D\}$. In addition, the crossover rate $CR \in [0,1]$ is set by user.

3) Selection: In a competition way relying on the fitness value, we select one between the target individual and trail individual, which will survive and get into the next gengeration. For the minimization problem, the selection rule is expressed as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if\ \ f(U_{i,G+1}) < f(X_{i,G}) \\ X_{i,G} & otherwise \end{cases} \tag{12}$$

In our modified algorithm, F and CR are the control parameters and affect the performance of the algorithm largely.

B. Estimation of Distribution Algorithm

Estimation of Distribution Algorithm is directly describe the trend of the evolution of the entire population, and is the mathematical model of biological evolution based on the "macro" level. Unlike other EAs, there are no mutation and crossover operations in EDA. So, it can maintain the global information greatly in the search process and have the strong exploration capability. EDA generates new population according to the sampling from probabilistic model learned from the current population. EDA works as follows iteratively:

1) Selection: Selection aims to obtain a set of individuals called as selected population that will be learned and modeled by a secletion method. It is expected that the selected individuals can describe the solution space accurately and comprehensively. The widely-used method is the truncation selection. In the trunction selection, the best individuals are selected relying on their objective function values.

2) Modeling: The probabilistic model is constructed based on the statistical information extracted from the selected individuals using a learning method.

3) Sampling: When a sampling method is employed, the new population called as sampling population is generated from the constructed probabilistic model. Therefore, the sampling method is denpendent on the used learning method.

4) Replacement: In every generation, to obtain the offspring population, a replacement method is employed to combine the individuals generated in the sampled population with individuals from previous generations. Especially, the individuals generated randomly according to the uniform distribution in the search space can also be selected into the offspring population.

C. The Proposed Algorithm

A novel hybrid DE-EDA algorithm for dynamic optimization problems is proposed. In our algorithm, DE and EDA are combined in the form of series according to incorporating the mutation and crossover schemes of DE algorithm. It can make use of not only the global information but also the local information sufficiently. Meanwhile, the empirical copula-based probabilistic model is used [18, 19, 25]. The mutation scheme is modified to increase the diversity of population. Fig. 1 shows the empirical copula-based DE-EDA algorithm.

The detailed operations of the proposed algorithm are presented as follows:

1) DE-EDA offspring generation scheme: In every generation, a offsping population is generated by employing the proposed scheme. Owing to having strong exploring capability, EDA always guides its search to a area in which the optimal solution has the opportunity existing by sampling new population from a probabilistic model. So we firstly generate a new populaion by conducting EDA operation once. After that, the mutation, crossover and selection schemes of DE are used to generate the offsping population. The optimal solution can be located easily because of the exploiting capability of DE. This is equivalent to a local search. In the generation G , the proposed DE-EDA offspring generation scheme works as table Fig. 2, where α is the selection rate in trunction selection method we used in our algorithm, and when the individual is evaluated, the standard Runge-Kutta method is employed to calculate the state variable.

2) Modeling and sampling scheme based on empirical Copula: We cosider the modeling and sampling method in [17,18,25]. A Copula $C(u)$ is a multivariate distribution function for the D-dimensional vector $u = (u_1, u_2, \ldots, u_D)$ on the unit D-cube $[0,1]^D$. For a D-dimensional random vector $x = (x_1, x_2, \ldots, x_D)$, its joint distribution function is

$F(x)$ with the marginal distribution $F_i(x_i)$, $i=1,2,\ldots,D$. In [18], the equality was set up as follows:

$$C(u_1,\ldots,u_D)=F(F_1^{-1}(u_1),\ldots,F_D^{-1}(u_D)) \tag{13}$$

So the method of generating a random vector $x=(x_1,x_2,\ldots,x_D)$ subject to the distribution $F(x)$ consists in two steps. First a random vector $u=(u_1,u_2,\ldots,u_D)$ is generated with the copula. Then the $x=(x_1,x_2,\ldots,x_D)$ is generated according to calculate the inverse function of the marginal distribution of each variable, the expression can be written as $x_i=F_i^{-1}(u_i)$, $i=1,2,\ldots,D$. In our algorithm, the sample individuals are generated using the method presented in [18].

3) Modified mutation scheme: To increase the diversity of the population, we modify the mutation scheme. In our modified DE, we add a random factor R subject to uniform distribution to increase the diversity and $R\in[0,1]$, so the mutation individual can be rewritten as follows:

$$\begin{aligned}V_{i,G} &= X_{Best,G}+F*(X_{r1,G}-X_{r2,G})\\ &\quad +(F+R)*(X_{r3,G}-X_{r4,G})\end{aligned} \tag{14}$$

Step 1: Discrete the time $[t_0,t_f]$ into D stages

Step 2: Initialize a D-dimension randomly distributed population X_0 of size N within the feasible search space

Step 3: Set the control parameter vector $[\alpha,F,C_r]$

Step 4: Set generation number $G=0$

WHILE stopping criteria not met

Step 5: Generate a offspring population X_G according to the DE-EDA offspring generation Scheme

Step 6: Increment generation number $G=G+1$

END WHILE

Fig. 1. Pseudo-code of the empirical copula-based DE-EDA algorithm

Step 1: Select a population X_G^{Select} of size $\alpha*N$ from the current population X_G of size N employing the truncation selection

Step 2: Construct a probabilistic model p_m based on the empirical Copula using the population X_G^{Select}

Step 3: Sample a population X_G^{Sample} of size $(1-\alpha)*N$ from the model p_m

Step 4: Evaluate each individual of the population X_G^{Sample}

Step 5: Create a new population X_G^{New} by combining X_G^{Select} with X_G^{Sample}

Step 6: DO FOR $i=1$ to N

Step 6.1: Generate mutation vector $v_{i,G}=[v_{1,i,G},v_{2,i,G},\ldots,v_{D,i,G}]$ using scaling factor F, random factor R and the modified strategy from X_G^{New}

Step 6.2: Employ crossover method described below and crossover probability C_r to obtain trial vector $u_{i,G}=[u_{1,i,G},u_{2,i,G},\ldots,u_{D,i,G}]$

END FOR

Step 7: Evaluate each individual of trial vector population U_G^{New}

Step 8: Generate the offspring population X_{G+1} using the selection method described below

Fig. 2. Pseudo-code of the DE-EDA offspring generation scheme

IV. COMPUTATIONAL EXPERIMENTS

To measure the performance of our algorithm, the hybrid DE-EDA algorithm was applied to several commonly used dynamic optimization problems and was compared with the DE algorithm and the EDA algorithm separately, as well as the best solutions in the literature in this section.

A. The Test Problems

The problems with varying levels of difficulty are chosen from the literature typically. The first eight are the optimal control problems selected from [20-24]. The last one is the parameter estimation problem. The description of the test problems is presented in TABLE I. The first four J_1, J_2, J_3 and J_4 are the minimum problems. The J_5, J_6, J_7 and J_8 are the maximum problems. Specially, J_5 is a batch reactor problem having the consecutive reactions $A\to B\to C$, J_6 is modeled according to a tubular rector in which the parallel reactions is $A\to B, A\to C$, J_7 is considered by a catalytic plug flow rector with the following reactions $A\leftrightarrow B\to C$ and J_8 aims to optimize the feed rate of Lee-Ramirez bioreactor.

TABLE I. DESCRIPTION OF TEST PROBLEMS

Ser. No.	Function description
J_1	$\min_{u(t)} J=x_2(t_f)$ $s.t.\ \ \dot{x}_1=u, \quad x_1(0)=1$ $\dot{x}_2=x_1^2+u^2, \quad x_2(0)=0$ $-10\le u\le 10$ $t_f=1$

J_2 $\min_{u(t)} J = x_2(t_f)$

$$s.t.\ \ \dot{x}_1 = u, \quad x_1(0) = 1$$
$$\dot{x}_2 = x_1^2 + u^2, \quad x_2(0) = 0$$
$$x_1(1) = 1$$
$$-10 \le u \le 10$$
$$t_f = 1$$

J_3 $\min_{u(t)} J = x_4(t_f)$

$$s.t.\ \ \dot{x}_1 = x_2$$
$$\dot{x}_2 = -x_3 u + 16t - 8$$
$$\dot{x}_3 = u,\ x_3(0) = -\sqrt{5}$$
$$\dot{x}_4 = x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1 x_3 u^2)^2$$
$$x(0) = [0\ \ -1\ \ -\sqrt{5}\ \ 0]^T$$
$$-4 \le u \le 10$$
$$t_f = 1$$

J_4 $\min_{u(t)} J = \int_0^1 (x_1^2 + x_2^2 + 0.005u^2)dt$

$$s.t.\ \ \dot{x}_1 = x_2, \quad x_1(0) = 0$$
$$\dot{x}_2 = -x_2 + u, \quad x_2(0) = -1$$
$$x_2 - 8(t-0.5)^2 + 0.5 \le 0, t \in [0,1]$$
$$-10 \le u \le 10$$
$$t_f = 1$$

J_5 $\max_{T} J = x_2(t_f)$

$$s.t.\ \ \dot{x}_1 = -k_1 x_1^2, \quad x_1(0) = 1$$
$$\dot{x}_2 = k_1 x_1^2 - k_2 x_2, \quad x_2(0) = 0$$
$$0 = k_1 - 4000e^{(-2500/T)}$$
$$0 = k_2 - 620000e^{(-5000/T)}$$
$$298 \le T \le 398$$
$$t_f = 1$$

J_6 $\max_{u(t)} J = x_2(t_f)$

$$s.t.\ \ \dot{x}_1 = -(u + 0.5u^2)x_1, \quad x_1(0) = 1$$
$$\dot{x}_2 = u x_1, \quad x_2(0) = 0$$
$$0 \le u \le 5$$
$$t_f = 1$$

J_7 $\max_{u(t)} J = 1 - x_1(t_f) - x_2(t_f)$

$$s.t.\ \ \dot{x}_1 = u(10x_2 - x_1), \quad x_1(0) = 1$$
$$\dot{x}_2 = -u(10x_2 - x_1) - (1-u)x_2, \quad x_2(0) = 0$$
$$0 \le u \le 1$$
$$t_f = 12$$

J_8 $\max_{u(t)} J = x_4(t_f)x_1(t_f) - Q\int_0^{t_f} u_2 dt$

$$s.t.\ \ \dot{x}_1 = u_1 + u_2$$
$$\dot{x}_2 = \mu x_2 - \frac{u_1 + u_2}{x_1} x_2$$
$$\dot{x}_3 = \frac{u_1}{x_1} C_{nf} - \frac{u_1 + u_2}{x_1} x_3 - Y^{-1}\mu x_2$$
$$\dot{x}_4 = R_{fp} x_2 - \frac{u_1 + u_2}{x_1} x_4$$
$$\dot{x}_5 = \frac{u_2}{x_1} C_{if} - \frac{u_1 + u_2}{x_1} x_5$$
$$\dot{x}_6 = -k_1 x_6$$
$$\dot{x}_7 = k_2(1 - x_7)$$
$$\mu = \frac{0.407x_3 \times (x_6 + x_7(\frac{0.22}{0.22 + x_5}))}{0.108 + x_3 + x_3^2/14814.8}$$
$$R_{fp} = \frac{0.095x_3}{0.108 + x_3 + x_3^2/14814.8} \times (\frac{0.0005 + x_5}{0.022 + x_5})$$
$$C_{nf} = 100.0$$
$$C_{if} = 4.0$$
$$k_1 = k_2 = \frac{0.09x_5}{0.034 + x_5}$$
$$Y = 0.51$$
$$Q = 5$$
$$x(0) = [1\ \ 0.1\ \ 40\ \ 0\ \ 0\ \ 1\ \ 0]^T$$
$$0 \le u_1 \le 0.01$$
$$0 \le u_2 \le 0.01$$
$$t_f = 10(h)$$

J_9 $\min_{p} J = \sum_{i=1,2,3,5} (x_1(t_i) - x_1^m(t_i))^2$

$$s.t.\ \ \dot{x}_1 = x_2$$
$$\dot{x}_2 = 1 - 2x_2 - x_1$$
$$-1.5 \le p_1 \le 1.5$$
$$-1.5 \le p_2 \le 1.5$$
$$x(0) = [p_1, p_2]^T$$
$$x_1^m(t_i) = [0.264\ \ 0.594\ \ 0.801\ \ 0.959]$$
$$i = 1,2,3,5$$
$$t_f = 6$$

B. Parameter Setup

The parameters of DE-EDA are given in TABLE II. For the first eight problems, the number of time stages D was set as 80. In the last problem, D is unnecessary. In DE, the values of F and C_r is the same as in DE-EDA. In a similar way, the value of α is equivalent to the value in EDA.

TABLE II. PARAMETERS OF THE DE-EDA

Parameter	Value
N	200
F	0.5
C_r	0.1
α	0.2
D	80

For each problem, 30 runs of each DE, EDA, and DE-EDA algorithms were implemented in order to avoid the influence of fluctuation caused by the random number generator on the quality of the solutions. And in each run the three algorithms were executed for 1500 iterations. The algorithms were completed in C# language in the platform of Microsoft Visual Studio 2005. And the experimental results of this study were obtained utilizing a Lenovo computer with the configuration of Intel(R) Core(TM) i5-2400/3.10GHz/RAM 4.00GB.

C. Experimental Results

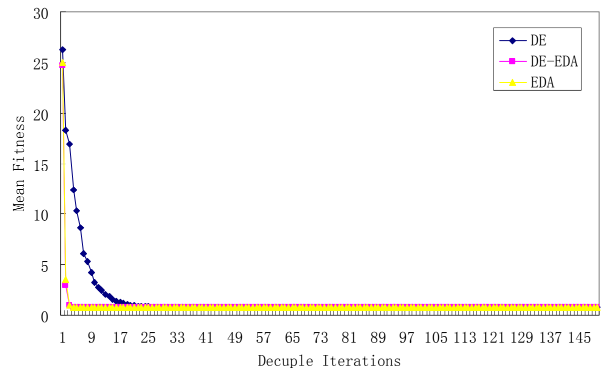
1) The optimal control porblems: For the three algorithms, the average(mean) function value is calculated from the 30 simulation runs and then compared. The experimental results are shown in TABLE III. Results highlighted in bold indicate that the solutions of the functions obtained by our algorithm are superior to the best solutions from literature and other two algorithms. To have an insight into the difference of the three algorithms, convergence graphs for the eight optimal control problems are sketched in Fig. 3. The evolution procedures of the test problems from J_1 to J_8 are labeled from (a) to (h). It is observed that the average best objective values found in the DE-EDA and DE decrease faster than those found in EDA algorithms and reach near the optimal solution within 100 generations expect Fig. 3(d). Though the EDA converges quickly, the quality of solutions is inferior to that of DE-EDA.

TABLE III. MEAN FUNCTION VALUE OVER 30 RUNS

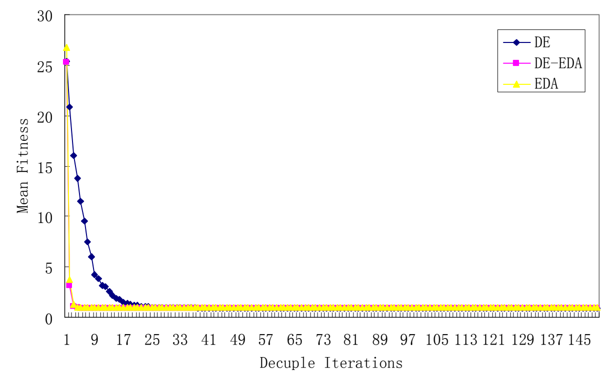
Function	DE-EDA	EDA	DE	Best From Literature
J_1	0.7616018	0.76996	0.7616018	0.7615959
J_2	0.9242703	0.93844	0.9242712	0.9242346
J_3	**0.1192927**	0.12386	0.1195812	0.1202688
J_4	0.1798737	0.18598	0.1919130	0.1701564
J_5	**0.6107762**	0.61038	0.6107761	0.6107750
J_6	0.5735279	0.57197	0.5735279	0.57353
J_7	**0.4771443**	0.47670	0.4770933	0.476946
J_8	**0.8164747**	0.81641	0.8164728	0.816400

2) A specific case study: In order to make the proposed algorithm more easy to be understood, the process of solving the dynamic optimization problem is given in the following. The problem J_1 is considered as a case. The algorithm starts with a 80-dimension population of size 200 in the feasible region $[-10,10]$ of the variable u. And the parametes F , C_r and α are set as 0.5, 0.1 and 0.2 separately. Then the proposed DE-EDA offspring generation scheme described in the section III is employed interatively to obtain the new generation. The algorithm ends with the iterations being equal to 1500. When the individual is evaluated, the standard Runge-Kutta method is employed to calculate the state variables x_1 and x_2 . And the last value of x_2 is served as the fitness of the function J_1 .

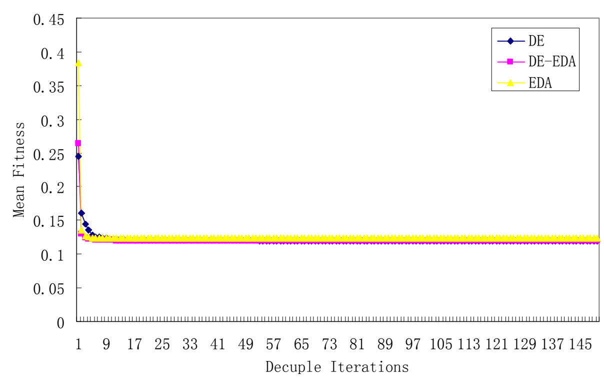
3) The parameter estimation porblem: For the problem J_9 , the cost function is defined as the sum of squares of deviations between the model and measured outputs. Finding the initial conditions is our task Fig. 4 shows the estimated values of the state x_1 and x_2 in the full time interal. And the comparion of estimated and measured state trajectory for state x_1 is presented. The initial values of x_1 and x_2 obtained by our algorithm are -0.0011217 and 0.0016245 respectively.



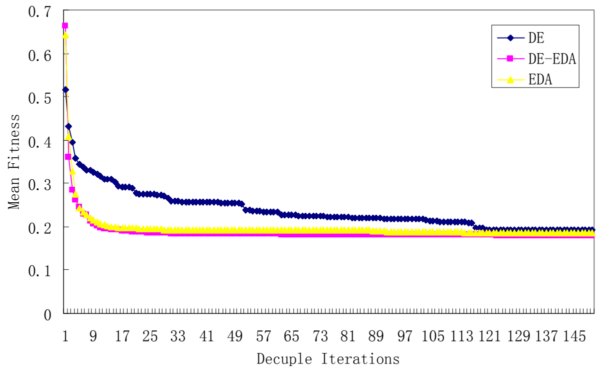
(a)



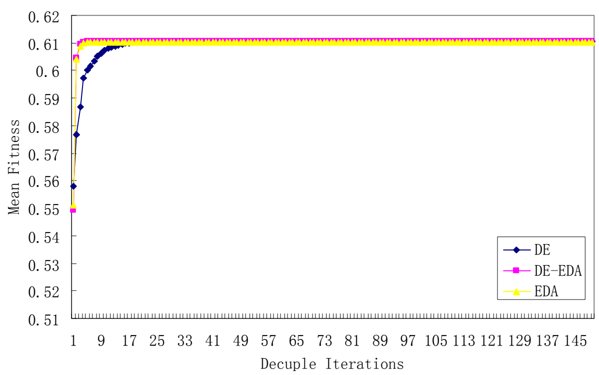
(b)



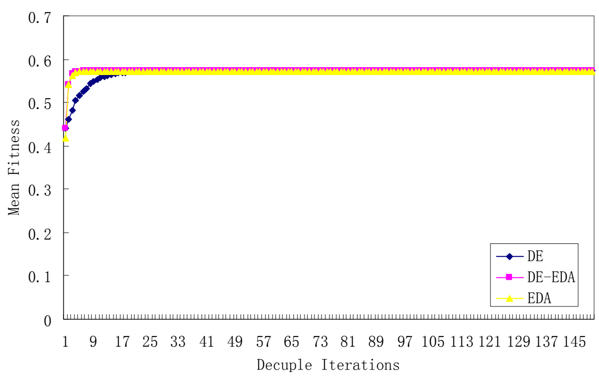
(c)



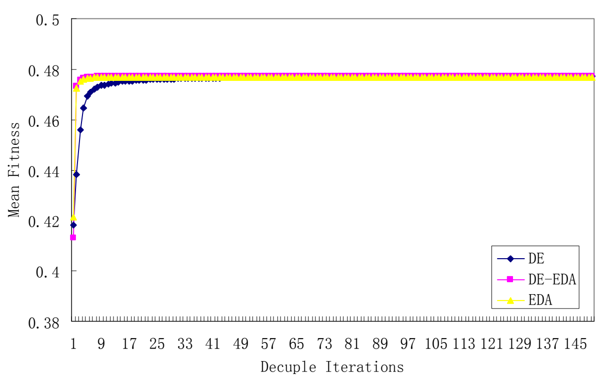
(d)



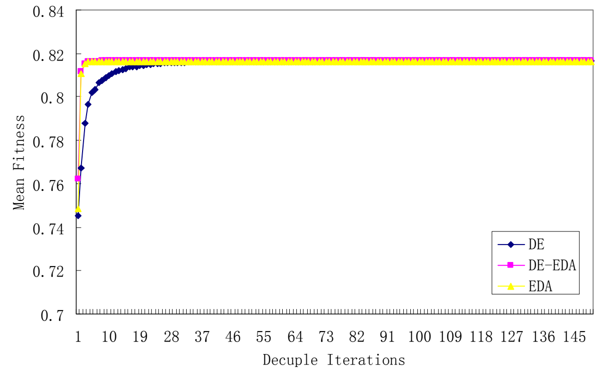
(e)



(f)



(g)



(h)

Fig. 3. Convergence graph for the ten benchmark functions

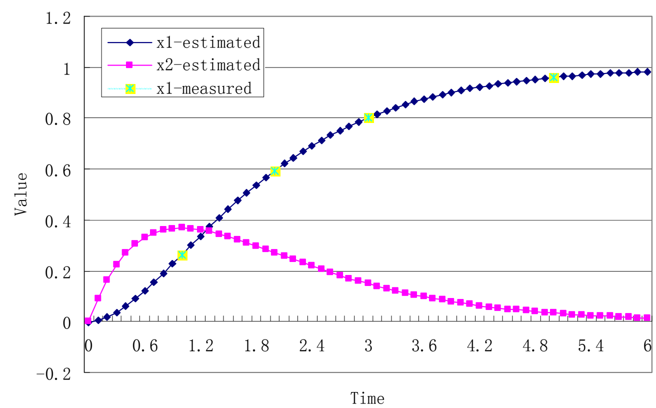


Fig. 4. Estimated and measured state trajectory for state x_1 and x_2

V. CONCLUSION

In this paper, a novel DE-EDA algorithm is proposed and evaluated for solving the dynamic optimization problems. The hybrid scheme based on DE and EDA is designed to generate the offspring population. It can incorporate the local information gained by DE operator and the global information collected by modeling and sampling scheme based on empirical Copula together. Using the DE-EDA, the population can reach a promising area in which the optimal solution is located speedily.

DE-EDA is applied to solve eight optimal control problems and one parameter estimation problem. In the experiments we have compared DE-EDA with the version of DE/best/2/bin and the EDA. The experimental results indicate that the proposed algorithm can obtain better solutions and converges more quickly than two other algorithms, even beat the best solutions from the literature for some given optimal problems. Our algorithm also can estimate the parameter of some control systems precisely.

In the future, we will design a scheme to prompt the speed of the modeling and sampling of the population.

REFERENCES

[1] H. C. Lim, Y. J. Tayeb, J. M. Modak, and P. Bonte, “Computational algorithms for optimal feed rates for a class of fed-batch fermentation: Numerical results for penicillin and cell mass production,” Biotechnology and Bioengineering, vol. 28, 1986, pp. 1408-1420.

[2] P. I. Barton, R. J. Allgor, W. F. Feehery, and S. Galan, “Dynamic optimization in a discontinuous world,” Industrial & Engineering Chemistry Research, vol. 37, 1998, pp. 966-981.

[3] R. Angira, A. Santosh. “Optimization of dynamic systems: A trigonometric differential evolution approach,” Computers&Chemical Engineering, vol. 31, 2007, pp. 1055-1063.

[4] J. A. Roubos, G. van Straten, A. J. B. van Boxtel, “An evolutionary strategy for fed-batch bioreactor optimization: concepts and performance,” Journal of Biotechnology, vol. 67, 1999, pp. 173-187.

[5] L. T. Biegler, “An overview of simultaneous strategies for dynamic optimization,” Chemical Engineering and Processing, vol. 46, 2007, pp.1043-1053.

[6] R. Angira, A. Santosh, “Optimization of dynamic systems: A trigonometric differential evolution approach,” Computers and Chemical Engineering, vol. 31, 2007, pp. 1055-1063.

[7] A. Cervantes, L. Biegler, “Large-scale DAE optimization using a simultaneous NLP formulation”, AIChE Journal , vol. 44 (5) , 1998, pp. 1038-1050.

[8] H. Bock, K. Plitt, “A multiple shooting algorithm for direct solution of optimal control problems,” Proceedings of the 9th IFAC World Congress Budapest, Pergamon, Oxford, 1984.

[9] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations,” Journal of Process Control, vol. 12, 2002, pp. 577-585.

[10] F. S. Wang, J. P. CHiou, “Optimal control and opotimal time location problems of differential algebraic systems by differential evolution,” Industrial and Engineering Chemistry Research, vol. 36, 1997, pp. 5348-5357.

[11] J. Rajesh, K. Gupta, H. Kusumakar, V. K. Jayaraman, and B. D. Kulkarni, “Dynamic optimization of chemical processes using ant colony framework,” Computers and Chemistry, vol. 25, 2001, pp. 583-595.

[12] S. R. Upreti, “A new robust technique for optimal control of chemical engineering processes,” Computers & Chemical Engineering, vol. 28, 2004, pp. 1325-1336.

[13] J. Lepagnot, A. Nakib, H. Oulhadj and P. Siarry, “A multiple local search algorithm for continuous dynamic optimization,” J. Heuristics, vol.19, 2013, pp. 35-76.

[14] J. Lepagnot, A. Nakib, H. Oulhadj and P. Siarry, “A New Multiagent Algorithm for Dynamic Continuous Optimization,” Int. J. of Applied Metaheuristic Computing, vol. 1, 2010, pp. 16-38.

[15] H. Mühlenbein, G. Paaß, “From recombination of genes to the estimation of distribution,” Binary parameters. Lecture notes in computer science 1411: parallel problem solving from nature, PPSN, 1996, pp. 178-187.

[16] J. Y. Sun, Q.F. Zhang, and E. P. K. Tsang, “DE/EDA: A New Evolutionary Algorithm for Global Optimization,” Information Sciences, vol. 169, 2005, pp. 249-262.

[17] R. Storn and K. Price, “Differential Evolution-A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” Journal of Global Optimization, vol. 11, 1997, pp. 341-359.

[18] J. Strelen, F. Nassaj, “Analysis and generation of random vectors with copulas,” Proceeding of the 2007 Winter Simulation Conference (WSC2007), vol. 26(9), 2007, pp. 488-496.

[19] L. F. Wang, J. C. Zeng, and Y. Hong, “Estimation of distribution algorithm modeling and sampling by means of Copula,” Control and Decision, 2011, pp. 1335-1342.

[20] R. Luus, “Application of iterative dynamic to state constrained optimal control problems,” Hung. J. Ind. Chem., vol. 19, 1991, pp. 245-254.

[21] D. Jacobson and M. Lele, “A transformation technique for optimal control problemswith a state variable inequality constraint,” IEEE Trans. Automatic Control, vol. 5, 1969, pp. 457-464.

[22] R. Luus, “Optimal control by dynamic programming using systematic reduction in grid size,” Int. J. Control, vol. 51, 1990, pp. 995-1013.

[23] S. A. Dadebo and K. B. McAuley, “Dynamic optimization of constrained chemical engi-neering problems using dinamic programming,” Computers chem. Engng., vol. 19, 1995, pp. 513-525.

[24] J. Lee, W. F. Ramirez, “Optimal fed-batch control of induced foreign protein production by recombinant bacteria,” AIChE J, vol. 40, 1994, pp. 899-907.

[25] R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E. R. Villa-Diharce, “Using Copulas in Estimation of Distribution Algorithms,” MICAI '09 Proceedings of the 8th Mexican International Conference on Artificial Intelligence, 2009, pp. 658-668.