Proceeding of the 11th World Congress on Intelligent Control and Automation
Shenyang, China, June 29 - July 4 2014

Dynamic Optimization of Chemical Engineering Problems Using Affinity Propagation Based Estimation of Distribution Algorithm

Na Luo, Wei Feng, Xiaoqiang Wang, Feng Qian*
Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education
East China University of Science and Technology
Shanghai,China
fqian@ ecust.edu.cn

***Abstract*—Dynamic optimization has attracted much attention for its wide applications in engineering problems. However, it is still a challenge for high nonlinear, multi-dimensional and multimodal problems. Estimation of Distribution Algorithm was proposed in which probabilistic models extracted relevant features of the complex search space and then generated new individuals during optimization. In order to decrease the dependences among control variables in dynamic optimization, affinity propagation was applied to cluster the individuals in evolutionary iterations. In each cluster, the probabilistic density function of Gaussian mixture model refined the promising spaces with high quality solutions and avoided the random combination of different control variables. To evaluate the performance of the new approach, three dynamic optimization problems of chemical process are used as cases comparing with three state-of-the-art global optimization methods. The results showed that the new approach could achieve the best solution in most cases with less computational effort and higher efficiency.**

Keywords—Dynamic optimization; Estimation of distribution algorithm; Chemical process; affinity propagation; Gaussian mixture model.

I. INTRODUCTION

Dynamic Optimization Problems (DOPs) are widely spread in chemical engineering applications. In general, it is typical to find a predefined performance index to subject to certain specification over a time interval. The DOP can be described as follows:

$$\min\ J(u)=\Phi\left[x(t_f)\right]+\int_{t=0}^{t=t_f}\Psi\left[x(t),u(t),t\right]dt \qquad (1)$$

$$\text{s.t.}\quad \varphi\left(x(t),u(t),t\right)=0$$

$$\dot{x}(t)=f\left(x(t),u(t),t\right)$$

where x and u are called state variables and control variables respectively. $J(u)$ is defined as the objective function. Φ and Ψ are the general function relationship and the function vector separately. t_f is the end time. The control variables $u(t)$ and the state variables $x(t)$ as well as their derivatives with respect to time $\dot{x}(t)$ appear continuous in this problem. This kind of problem is known as Differential-Algebraic Equation (DAE) problem, which consists of differential equations that describe the dynamic behavior of the system and algebraic equations that ensure some relations. A variety of approaches have been studied in order to obtain the optimal solution of the DAE problems.

These approaches can be grouped into two general classes: the indirect and direct approach. The indirect or variational approaches obtain the optimality using the first order necessary conditions in Pontryagin's Maximum Principle. A review of the indirect approaches can be found in [1]. Different from indirect methods, direct approaches convert the original problem into a finite-dimensional one, usually a nonlinear programming problem (NLP). Usually, these NLPs can be solved by variations of Successive Quadratic Programming (SQP). Biegler[2] proposed an efficient reduced space SQP approach for DOPs in which the number of state variables is much larger than the number of control variables. Dynamic programming was also proved feasible[3] in DOPs. A systematic backward search method was used to find the optimal path after time and control variables were discretized to a predefined number of values. However, when the number of discritization became large, computation increases rapidly. Iterative dynamic programming[4] was proposed to avoid this difficulty. Using coarse grid points and region-reduction strategy, computation efficiency promoted and numerical accuracy increased. But when state variables were also discretized, it was still troublesome.

Comparing with dynamic programming, it was unnecessary to discretize states variables when evolutionary algorithms were applied in DOPs. Furthermore, control profile at all time stages could be optimized simultaneously. Genetic Algorithm (GA) was the first to solve DOPs. Pham[5] applied GA in DOPs by using several novel reproductive operators including shift, smoothing, extrapolation and swapping. Modified differential evolution algorithm was also tried by Lee[6] through employing a local search to enhance the computational efficiency and modifying heuristic constraints to systematically reduce the size of the search space. Angira[7] applied trigonometric differential evolution algorithm for solving DOPs to enhance convergence speed. Simulated annealing algorithm was also applied[8] in determining operation policies of a binary distillation[9]. As a low computational algorithm, ant colony algorithm had a potential for DOPs[10]. Zhang[11] developed iterative ant-colony algorithm to gradually approximate the optimal control profile. Likewise, population-based probability distribution estimation was applied in tackling constrained multistage complex process DOPs with special solution migration and penalty assignment techniques integrated[12]. Recently, hybrid algorithms such as improved particle swarm optimization with successive quadratic programming[13] were proposed which balanced the speed of convergence and the ability of escaping local optimal.

In general, evolutionary algorithms have been widely used in DOPs.

Certainly, improvement on evolutionary algorithms should be taken when DOPs need to be solved efficiently and accurately. Pham[14] used statistical analysis to generate progressive step reduction and improved the efficiency of the algorithm. Region reduction strategy was chosen by Asgari[15] to obtain more accurate results. Liu[16] applied orthogonal collocation to decrease the combination of different control variables and encoded chromosome for improvement of search efficiency with soft constraint strategy. In short, the search space should be oriented to promising area without considering mutual dependence of control variables. Comparing with other evolutionary algorithms, Estimation of Distribution Algorithm (EDA)[17] is based on probability distribution estimation. In EDAs, relevant features of the search space are extracted using a probabilistic model, which is later employed to generate new points. In this way, the search is easily oriented to promising areas. As the key factor, many probability models are proposed for real-world problems. For the best performance, time-dependent dynamic problems require its own probability model construction method to direct the optimization course. In this paper, various strategies are investigated to limit the research area and an improved EDA is used to solve dynamic problems efficiently.

This paper is organized as follows. Section 2 reviews different kinds of EDAs proposed for solving general optimization problems. In section 3, an improved approach named AffEDA is illustrated for dynamic problems. To increase the independence of individual, affinity propagation is applied to cluster individuals. In each cluster, Gaussian mixture model is applied to construct the probabilistic model and generate new individuals, which avoids the random combination of different control variables. Section 4 presents three dynamic optimization cases for experiments and obtains results. The performance and computation efficiency are also analyzed comparing with other evolutionary algorithms. In the end, conclusions are drawn.

II. A GENERAL REVIEW OF AFFINITY PROPAGATION BASED ESTIMATION OF DISTRIBUTION ALGORITHM

EDA was originally proposed as a new evolutionary algorithm.. It evolves towards the promising area of the search space using the probabilistic and statistical theory. There are four main steps: selecting individuals, building probabilistic model, sampling from the model and replacing some individuals. Different probabilistic models are applied in EDAs. According to the probabilistic models, EDAs are categorized into three classes. The first class of EDAs uses a product of independent univariate probabilities, such as the probabilistic incremental learning (PBIL)[18], the compact genetic algorithm (CGA)[19] and the univariate marginal distribution algorithm (UMDA)[20], in which variables are assumed independent. In the second-class EDAs, the variables are disposed as bivariate dependences and the probabilistic model is developed based on Bayesian deduction. This class includes the mutual information maximization for input clustering (MIMIC)[21], the combining optimizers with mutual information trees algorithm (COMIT)[22] and the bivariate marginal distribution algorithm (BMDA)[23]. The third class which includes the extended compact genetic algorithm (ECGA)[24], the Bayesian optimization algorithm (BOA)[25] and the factorized distribution algorithm (FDA)[26] uses multivariate dependent Bayesian network to build the probability model. In continuous domain, Gaussian model is widely used for its simplicity. Based on these combinatory optimization algorithms, UMDAC[27], PBILC[28], MIMICC[27], EGNAee[27] are the extensions of UMDA, PBIC, MIMIC, EGNA to continuous domain based on Gaussian network.

Estimation of Distribution Algorithm based on Affinity Propagation (AffEDA) was first proposed by Santana[29]. This method learns marginal product models using affinity propagation and takes the matrix of mutual information between the variables as the similarity measure. Through grouping the variables in non-overlapping sets, strong interacting variables are contained in the same set. A general outline of AffEDA is as follows:

1) Initialize M individuals randomly and evaluate the fitnesses;

2) Select N (N<M) individuals according to a selection method and save some as the elite individuals;

3) Compute the mutual information between every pair of variables;

4) Apply Affinity clustering algorithm to obtain a marginal product model $p(x,t)$;

5) Sample M individuals according to the distribution $p(x,t)$;

6) Partially replace some new individuals by the elite individuals.

7) Stop if some stopping criterion is reached, else go to step 2.

AffEDA solves a given optimization problem by evolving a population of individuals towards promising zones of the search space. Such an evolution is mainly based on iteration between two steps: Select fit individuals from the current population, and combine the selected individuals in order to create an offspring population and partially replace the current one. For different problems, the interactions of variables are made clear using affinity clustering. By testing some benchmark functions, the AffEDA performs well. But for dynamic optimization problems, it is not so easy to find the solutions as usual.

III. AFFEDA FOR DYNAMIC OPTIMIZATION

For dynamic optimization, AffEDA needs to be modified to handle time-dependent dynamic problems. The main frame of AffEDA for dynamic problems includes: (i) discretizing time interval; (ii) constructing the probabilistic model and searching optimal control profile of the discrete system; (iii) returning to step (ii) for next iteration until reaching convergence. In all of these three steps, the important modification is the new construction of the probabilistic model.

3.1 Discretization of time horizon

To solve the dynamic optimization problems directly, the time interval is uniformly divided into n stages. So the i th time stage is $[t_i, t_{i+1}]$ where $t_1 = 0$ and $t_{n+1} = t_f$. Control variables are discredited into a profile according to the time interval. Assumed that there are m control variables, the optimization objective is to identify the optimal values from $m \times n$ discretized system inputs. It is obvious that if all possible inputs are calculated, the computation complexity is

large. So the relationship of decision variables should be considered to decrease the computation.

In order to improve the performance, it is feasible to reduce the search region during the iteration of the optimization. Different from general formula to decrease the limit of variables, AffEDA determines the range of the control profile using the probabilistic model. AFFEDA generates a population of complete solutions before any solution quality evaluation and the probabilistic model construction. And the inputs in all time stages are actually simultaneously optimized. In the next iteration, the individuals will be used for constructing the probabilistic model. Using AffEDA, a route of a single variable u with n time intervals is illustrated in Fig.1.

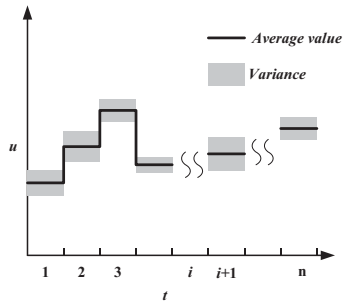

Fig. 1. Discretization of time interval

In Fig.1, the profile has n elements, each of which represents a feasible control strategy at a time stage. Around the heavy line is the variance of the control profile. That represents the range of the control profile in which the control profile has some probability to choose the value. Different from the other algorithms, the control profile is not just a determined line but a line with variance. In the next section, cluster method is used to make the algorithm concentrate on the mean value of the variables and decrease the computation complexity.

3.2 Probabilistic model construction

In order to construct the probabilistic model of the individuals, two questions should be considered. One is the selection method of the samples. The other question is how to construct the probabilistic model. In this paper, learning the mutual information of the probabilistic model is by a clustering method known as affinity propagation which referred the method proposed by Santana [29]. As the probabilistic model, Gaussian Mixture model is chosen.

- 3.2.1 Selection scheme

The initialization of the population is usually random. From the generated population, selection is performed to choose promising individuals. There are many selection methods, such as truncation, tournament selection or roulette wheel selection. During the selection course, the determination of the selection size is important. With small selection size, promising points may be excluded. On the contrary, the large selection size may increase the learning overhead.

Generally, the truncation selection is used. In truncation selection, individuals are sorted according to their fitness. Only a certain percentage of individuals are selected with which probability model is constructed. The parameter for truncation selection is the truncation ratio which indicates the proportion of the population to be selected. The ratio takes values ranging from 0.1-0.5. Individuals below the truncation threshold do not produce offspring.

- 3.2.2 Clustering with affinity propagation

Affinity propagation is a new developed clustering method which is proposed by Frey[30]. Different from the popular k -centers clustering technique, the affinity propagation method recursively transmits real-valued messages along edges of the network and is not sensitive to initial selection of exemplars. In affinity propagation, there are three definitions: similarity $s(i,k)$, responsibility $r(i,k)$ and availability $a(i,k)$.

$$s(i,k) = -\|x_i - x_k\| \tag{2}$$

$$r(i,k) \leftarrow s(i,k) - \max_{k',s.t.k'\neq k} \{a(i,k') + s(i,k')\} \tag{3}$$

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i',s.t.i'\notin\{i,k\}} \max\{0, r(i',k)\}\right\} \tag{4}$$

The similarity $s(i,k)$ indicates how well the data point with index k is suited to be the exemplar for data point i. When the goal is to minimize the squared error, each similarity is set to a negative squared error (Euclidean distance). The responsibility $r(i,k)$, sent from data point i to candidate the exemplar point k, reflects the accumulated evidence for how the well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. The availability $a(i,k)$, sent from the candidate exemplar point k to point i, reflects the accumulated evidence for how appropriate it would be for point i to choose the point k as its exemplar, taking into account the support from other points that the point k should be an exemplar.

In the first iteration, because the availabilities are zero, $r(i,k)$ is set to the input similarity between point i and point k as its exemplar, minus the largest of the similarities between point i and other candidate exemplars. This competitive update is data-driven and does not take into account how many other points favor each candidate exemplar. In later iterations, when some points are effectively assigned to other exemplars, their availabilities will drop below zero as prescribed by the update rule below. These negative availabilities will decrease the effective values of some of the input similarities $s(i,k')$ in the above rule, removing the corresponding candidate exemplars from competition. For $k = i$, the responsibility $r(k,k)$ is set to the input preference that the point k be chosen as an exemplar, $s(k,k)$, minus the largest of the similarities between point i and all other candidate exemplars. This self-responsibility reflects accumulated evidence that the point k is an exemplar, based on its input preference tempered by how ill-suited it is to be assigned to another exemplar.

- 3.2.3 Gaussian Mixture Model

A probabilistic model needs to be constructed each iteration. The model should be able to estimate the probability distribution of promising solutions so that the new individuals have a potential to be better ones. Further, the model should be constructed based on the solutions in the current iteration, which can be treated as a sample drawn from an unknown probability distribution. In order to construct a model for dynamic problems, Gaussian Mixture Model (GMM) is selected.

GMM is a parametric probability density function represented as a weighted sum of Gaussian component

densities[31]. With M component Gaussian densities, a Gaussian mixture model is given by the equation:

$$p(x|\lambda)=\sum_{i=1}^{M}\omega_i \mathrm{g}(x|\mu_i,\Sigma_i) \quad (5)$$

where x is a D-dimensional continuous-valued data vector, $\omega_i, i=1,\cdots,M$, are the mixture weights, and $\mathrm{g}(x|\mu_i,\Sigma_i)$, $i=1,\cdots,M$, are the component Gaussian densities. Each component density is a D-variate Gaussian function in the form:

$$\mathrm{g}(x|\mu_i,\Sigma_i)=\frac{1}{(2\pi)^{D/2}|\Sigma_i|^{D/2}}\exp\left\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\} \quad (6)$$

with mean vector μ_i and covariance matrix Σ_i. The mixture weights satisfy the constraint that $\sum_{i=1}^{M}\omega_i=1$. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation $\lambda=\{\omega_i,\mu_i,\Sigma_i\}$.

For clusters in section 3.2.2, the GMM mixture weights can be determined by:

$$\omega_i = C_i \Big/ \sum_{i=1}^{M} C_i \quad (7)$$

where C_i is the number of individuals in the i th cluster and M is the number of the cluster. In the i th cluster, the parameters μ_i,Σ_i are easy to be determined with expectation maximization algorithm.

3.3 Update of the population

Update of the population includes two steps: sampling and replacement. Sampling from the probabilistic model serves to generate the new population. In this paper, new population is sampled from Gaussian mixture model. In order to escape from the stagnation due to a very fast decrease of the variance, variances of the Gaussian mixture models are modified with a scale parameter. During the evolution, the sampling procedure is invoked at every generation, except the first one where seeding is applied.

In order to preserve the better solutions in the previous population and promote diversity in the current population, replacement is used. The elitism individual is selected from the previous populations proportional to fitness value and the new population is partially replaced by the elitism set according to the ranking of the fitness values.

3.4 Main procedure of AffEDA for dynamic optimization

In summary, given t_f, $u_{\max}$ and $u_{\min}$, the main procedure of AffEDA is described as below:

Step 1: Determine the number of iteration N, population size n_p, elitism size n_e and selection ratio τ

Step 2: Execute discretization of the time interval and randomly initialize the control variables according to section 3.1

Step 3: Select $\tau \times n_p$ individuals according to the fitness value, clustering these individuals using affinity propagation and construct GMM according to section 3.2 and 3.3; keep n_e elicit individuals in the current generation.

Step 4: Generate new samples from the GMM, and replace n_e worst samples using the elitism.

Step 5: If the stop criteria or the number of iteration N is reached, stop. Else return to Step 3 for the next iteration.

These algorithms are compared in terms of robustness, efficiency and scalability. For each problem, 10 runs of each algorithm are carried out. Considering the randomness of these heuristic methods, solutions are seen as equivalence and successful when they lie in an interval of range ±2%. The robustness is defined as the percentage of successful solutions along different runs. As for the efficiency of algorithms, it considers not only the number of successful solutions, but also the iterative number of generations. The efficiency index is defined as:

$$\mathrm{A}_{eff}=\frac{\text{Minimum Generations}}{\text{Average Generations}}\times\frac{\text{number of successful runs}}{\text{number of all runs}} \quad (8)$$

With this definition, the efficiency is given by the successful value of iteration numbers. Regarding scalability, it shows the convergence performance when increasing discretization levels with a reasonable increase of the computation effort.

IV. NUMERICAL EXPERIMENTS

4.1 Typical Cases

Three cases are selected to test the proposed method and the procedure. The first and seconcd case is from test problem 4 and 2 in [10] seperately, and the thrid case is from [32] named Park-Ramirez bioreactor.

4.2 Parameter setting

In order to test the performance of AffEDA, three process engineering dynamic optimization problems with different levels of complexity are considered in the present study. Control variables are discretized into n control stages according to the time interval and then integrated. Besides compared with the results coming from the references, three other global optimization methods, such as Genetic algorithm (GA), Particle Swarm Optimization (PSO) and Estimation of Distribution Algorithms using full multivariate Gaussian model (EDA-G) are applied as other comparisons. In these algorithms, the population size is set 100, and the initial population is generated uniformly. The algorithms run until the iteration is large than 100 or the weighted average change in the fitness function value over continuous five generations is less than 10^{-6} with the iteration larger than 50. Other specific parameters are listed in TABLE I.

TABLE I. PARAMETERS SETTING

GA	PSO	EDA-G	AffEDA
Migration Fraction: 0.2 Crossover Fraction: 0.8 Elite Count: 10	Initial inertia weight: 0.9 Final inertia weight: 0.4 acceleration constant 1: 2 acceleration constant 2: 2	Truncation ratio: 0.3 Elitism Population: 10	Truncation ratio: 0.3 Elite count: 10

4.3 Results and discussion

For Case 1, the time interval is discretized into 10, 20 and 50 number and control variable is discretized accordingly. With 25 runs for each algorithm, the results of AffEDA comparing with the results from references are shown in TABLE II. It is obvious that the global optimal found by AffEDA is as good as the results in the references. That means that AffEDA also has the strong search ability and can escape from local peaks for this case study.

In order to compare different algorithms in the same platform, the results of GA, PSO, EDA-G and AffEDA with different levels of discretization is presented in TABLE III. For this case, the best and worst solution using AffEDA is the same, which means that AffEDA can converge to the same optima every time. The robust value also illustrates this point. With the highest efficiency, AffEDA can obtain the optimal using the least iteration. With the increase of the time interval, better global optimal can be found using AffEDA while the search performance decreases using other algorithms such as GA. In summary, for this problem, the whole performance of AffEDA is superior to other algorithms.

TABLE II. COMPARISONS BETWEEN AFFEDA AND REFERENCES FOR CASE 1

Method	Time interval	Global optimal
ACA[10]		0.6105
IACA[11]	10	0.6100
IACA[11]	20	0.6104
Trigonometric Differential evolution[7]	10	0.6100
	20	0.6104
	50	0.6107
AffEDA	10	0.6101
	20	0.6105
	50	**0.6107**

TABLE III. RESULTS COMPARISON BETWEEN AFFEDA AND OTHER ALGORITHMS FOR CASE 1

		GA	PSO	EDA-G	AffEDA
n=10	best	0.6101	0.6098	0.6101	**0.6101**
	worst	0.6093	0.6001	0.6101	**0.6101**
	mean	0.6098	0.6082	0.6101	**0.6101**
	robust	100%	100%	100%	**100%**
	A_{eff}	98%	75%	100%	**100%**
n=20	best	0.6102	0.6063	0.6105	**0.6105**
	worst	0.6075	0.6006	0.6105	**0.6105**
	mean	0.6090	0.6035	0.6105	**0.6105**
	robust	100%	100%	100%	**100%**
	A_{eff}	98%	71%	100%	**100%**
n=50	best	0.6068	0.6023	0.6107	**0.6107**
	worst	0.6026	0.5564	0.6107	**0.6107**
	mean	0.6051	0.5971	0.6107	**0.6107**
	robust	100%	100%	100%	**100%**
	A_{eff}	96%	67%	100%	**100%**

For Case 2, the time interval is the same as that in case 1. The results of 10 different runs are shown in TABLE IV comparing with the global optimal from the literature. Clearly, AffEDA can find the smaller point for minimum problem and the obtained optimal is 5% less than the result using IACA.

Results of different algorithms used in case 2 is shown in TABLE V. In comparison, AffEDA can find the better point. Moreover, its robustness is close to 100% even if the discretization grows. While using GA, the robustness and efficiency decrease quickly with the increase of discretization. Because PSO searches the space more randomly, it is unsuitable for solving the dynamic problems without adaptation. Though EDA-G has the same structure with AffEDA, the clustering method and the proper probabilistic model direct AffEDA to search the space more effectively and efficiently.

In this paper, the case 3 is solved by discretizing 45 or 150 time intervals. TABLE VI shows the results using AffEDA comparing with results coming from literature. It is obvious that AffEDA is valid for complex dynamic problems. With the increase of time interval, AffEDA can find better solutions than other algorithms.

TABLE IV. COMPARISONS BETWEEN AFFEDA AND REFERENCES FOR CASE 2

Method	Time interval	Global optimal
ACA[10]		0.1290
IACA[11]	16	0.1253
AffEDA	10	0.1201
	20	0.1196
	50	**0.1199**

TABLE V. COMPARISONS BETWEEN AFFEDA AND OTHER ALGORITHMS FOR CASE 2

		GA	PSO	EDA-G	AffEDA
n=10	best	0.1206	0.1202	0.1202	**0.1201**
	worst	0.1321	0.1215	0.1213	**0.1215**
	mean	0.1237	0.1207	0.1205	**0.1205**
	robust	56%	100%	100%	**100%**
	A_{eff}	48%	106%	100%	**100%**
n=20	best	0.1207	0.1198	0.1198	**0.1196**
	worst	0.1344	0.1215	0.1205	**0.1202**
	mean	0.1240	0.1205	0.1200	**0.1201**
	robust	44%	100%	100%	**100%**
	A_{eff}	31%	92%	100%	**100%**
n=50	best	0.1268	0.1201	0.1199	**0.1199**
	worst	0.1953	0.1222	0.1202	**0.1201**
	mean	0.1383	0.1211	0.1201	**0.1200**
	robust	28%	100%	100%	**100%**
	A_{eff}	17%	81%	98%	**99%**

The results of AffEDA are compared with other algorithms shown in TABLE VII. For complex dynamic problems, evolutionary algorithm can't find better solutions with the increase of control variable discretizations, because the dimension of the variables increases than the performance of the algorithms. For AffEDA, with the increase of discretization, the robustness still keeps 100%. But the efficiency decreases, which means the more control variables cost more time for function fitness evaluations. In the whole the robustness, efficiency and scalability of AffEDA is superior to other algorithms.

TABLE VI. COMPARISONS BETWEEN AFFEDA AND REFERENCES FOR CASE 3

Method	Time interval	Global optimal
Particle Swarm Optimization [33]	20	32.4249-32.4541
Variance of Genetic Algorithm[32]	45	32.5928
Differential evolution [7]	45	32.6845
Trigonometric Differential evolution[7]	45	32.6843
AffEDA	45	**32.6690**
	150	**32.6685**

TABLE VII. COMPARISONS BETWEEN AFFEDA AND OTHER ALGORITHMS FOR CASE 3

		GA	PSO	EDA-G	AffEDA
n=45	best	32.4725	32.5360	32.6603	**32.6690**
	worst	31.1243	29.7593	32.5364	**32.5454**
	mean	31.9870	31.6352	32.5771	**32.6270**
	robust	88%	100%	100%	**100%**
	A_{eff}	77%	112%	92%	**94%**
n=150	best	32.6542	32.2353	32.6643	**32.6685**
	worst	31.1115	30.3731	32.5436	**32.5235**
	mean	32.2205	31.9377	32.6244	**32.6446**
	robust	60%	100%	100%	**100%**
	A_{eff}	39%	51%	78%	**67%**

In summary, by comparing among different evolutionary algorithms, the performance ranks AffEDA the first place regarding both solving problems and efficiency for the set of case studies considered in this work. Furthermore AffEDA obtained better results for increasing discretization levels practically all cases with a reasonable increase of the

computational effort. In contrast, all other methods experienced convergence difficulties when increasing the number of decision variables.

V. CONCLUSIONS

In this work, we presented a new probability distribution estimation based algorithm AffEDA for solving dynamic optimization problems. This novel approach introduces affinity propagation which clusters the populations during the evolution. For each cluster, Gaussian mixture model is used for constructing the probabilistic density functions. By comparing with other evolutionary algorithms, AffEDA refines the promising spaces containing high quality solutions. Based on these strategies, AffEDA speeds up the convergence to optimal solutions without performing too many redundant searches. To evaluate the performance of the new approach, three dynamic optimization problems of chemical process are used. For the sake of comparison, these problems were also optimized by several state-of-the-art global optimization methods. The results revealed that AffEDA could achieve the best solution in most cases with the least computational effort. In addition its performance was shown to scale up well with increasing discretization levels of the control variables, allowing the computation of more refined control profiles in reasonable computation times.

ACKNOWLEDGEMENT

Supported by Major State Basic Research Development Program of China (2012CB720500), National Natural Science Foundation of China (U1162202, 61222303), the Fundamental Research Funds for the Central Universities, Shanghai Municipal Natural Science Foundation(No.13ZR1411500) and Shanghai Leading Academic Discipline Project (B504).

REFERENCES

[1] Cervantes, A., and Biegler, L.T.: 'Optimization strategies for dynamic systems' (Kluwer Academic Publishers, 1999. 1999)

[2] Biegler, L.T., Cervantes, A.M., and Wachter, A.: 'Advances in simultaneous strategies for dynamic process optimization', Chem. Eng. Sci., 2002, 57, (4), pp. 575-593

[3] Dadebo, S.A., and McAuley, K.B.: 'Dynamic optimization of constrained chemical engineering problems using dynamic programming', Comput. Chem. Eng., 1995, 19, (5), pp. 513-525

[4] Chen, C.-L., Sun, D.-Y., and Chang, C.-Y.: 'Numerical solution of dynamic optimization problems with flexible inequality constraints by iterative dynamic programming', Fuzzy Sets and Systems, 2002, 127, (2), pp. 165-176

[5] Pham, Q.T.: 'Dynamic optimization of chemical engineering processes by an evolutionary method', Comput. Chem. Eng., 1998, 22, (7-8), pp. 1089-1097

[6] Lee, M.H., Han, C., and Chang, K.S.: 'Dynamic Optimization of a Continuous Polymer Reactor Using a Modified Differential Evolution Algorithm', Ind. Eng. Chem. Res., 1999, 38, (12), pp. 4825-4831

[7] Angira, R., and Santosh, A.: 'Optimization of dynamic systems: A trigonometric differential evolution approach', Comput. Chem. Eng., 2007, 31, (9), pp. 1055-1063

[8] Faber, R., Jockenhovel, T., and Tsatsaronis, G.: 'Dynamic optimization with simulated annealing', Comput. Chem. Eng., 2005, 29, (2), pp. 273-290

[9] Li, P., Lowe, K., Arellano-Garcia, H., and Wozny, G.: 'Integration of simulated annealing to a simulation tool for dynamic optimization of chemical processes', Chem. Eng. Process., 2000, 39, (4), pp. 357-363

[10] Rajesh, J., Gupta, K., Kusumakar, H.S., Jayaraman, V.K., and Kulkarni, B.D.: 'Dynamic optimization of chemical processes using ant colony framework', Computers & Chemistry, 2001, 25, (6), pp. 583-595

[11] Zhang, B., Chen, D., and Zhao, W.: 'Iterative ant-colony algorithm and its application to dynamic optimization of chemical process', Comput. Chem. Eng., 2005, 29, (10), pp. 2078-2086

[12] Xiao, J., Huang, Y., and Lou, H.H.: 'A probability distribution estimation based method for dynamic optimization', AlChE J., 2007, 53, (7), pp. 1805-1816

[13] Modares, H., and Sistani, M.B.N.: 'Solving nonlinear optimal control problems using a hybrid IPSO-SQP algorithm', Engineering Applications of Artificial Intelligence, 2011, 24, (3), pp. 476-484

[14] Pham, Q.T.: 'Using statistical analysis to tune an evolutionary algorithm for dynamic optimization with progressive step reduction', Comput. Chem. Eng., 2007, 31, (11), pp. 1475-1483

[15] Asgari, S.A., and Pishvaie, M.R.: 'Dynamic Optimization in Chemical Processes Using Region Reduction Strategy and Control Vector Parameterization with an Ant Colony Optimization Algorithm', Chemical Engineering & Technology, 2008, 31, (4), pp. 507-512

[16] Liu, Z., Du, W., Qi, R., and Qian, F.: 'Dynamic optimization in chemical processes using improved knowledge-based cultural algorithm', Huagong Xuebao/CIESC Journal, 2010, 61, (11), pp. 2889-2895

[17] Larrañaga, P., and Lozano, J.A.: 'Estimation of Distribution Algorithms : A New Tool for Evolutionary Computation' (Kluwer, 2002. 2002)

[18] Baluja, S.: 'Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and competitive Learning', in Editor (Ed.)^(Eds.): 'Book Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and competitive Learning' (Carnegie Mellon University, 1994, edn.), pp.

[19] Harik, G.R., Lobo, F.G., and Goldberg, D.E.: 'The compact genetic algorithm', Evolutionary Computation, IEEE Transactions on, 1999, 3, (4), pp. 287-297

[20] Mühlenbein, H., and Paaß, G.: 'From recombination of genes to the estimation of distributions I. Binary parameters', Lecture Notes in Computer Science, 1996, 1141, pp. 178-187

[21] Jeremy, S., De, B., Charles, L.I., and Viola, P.: 'MIMIC: finding optima by estimating probability densities'. Proc. Advances in Neural Information Processing Systems, Cambridge, MA1997 pp. Pages

[22] Baluja, S., and Davies, S.: 'Combining multiple optimization with optimal dependency trees', in Editor (Ed.)^(Eds.): 'Book Combining multiple optimization with optimal dependency trees' (Carnegie Mellon University, Pittsburgh, 1997, edn.), pp.

[23] Pelikan, M., and Mühlenbein, H.: 'The bivariate marginal distribution algorithm': 'Advances in soft computing: engineering design and manufacturing' (Springer-Verlag, 1999), pp. 521-535

[24] Harik, G., Lobo, F., and Sastry, K.: 'Linkage Learning via Probabilistic Modeling in the Extended Compact Genetic Algorithm (ECGA)': 'Studies in Computational Intelligence' (Springer-Verlag, 2006), pp. 39-61

[25] Pelikan, M., Goldberg, D.E., and Cantú-Paz, E.: 'BOA: The Bayesian optimization algorithm'. Proc. Genetic and Evolutionary Computation Conf.(GECCO-99)1999 pp. Pages

[26] Mühlenbein, H., and Mahnig, T.: 'Convergence theory and applications of the factorized distribution algorithm', Journal of Computing and Information Technology, 1999, 7, (1), pp. 19-32

[27] Larrañaga, P., Etxeberria, R., Lozano, J.A., and Peña, J.M.: 'Optimization in continuous domain by learning and simulation of Gaussian networks'. Proc. The Proceeding of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, Las Vegas, Nevada2000 pp. Pages

[28] Sebag, M., and Ducoulombier, A.: 'Extending Population-Based Incremental Learning to Continuous Search Spaces', Lecture Notes In Computer Science, 1998, 1498, pp. 418-427

[29] Santana, R., Larranaga, P., and Lozano, J.A.: 'Learning Factorizations in Estimation of Distribution Algorithms Using Affinity Propagation', Evolutionary Computation, 2010, 18, (4), pp. 515-546

[30] Frey, B.J.: 'Clustering by passing messages between data points', Science, 2007, 315, (5814), pp. 5

[31] Reynolds, D.A.: 'Gaussian Mixture Models', Encyclopedia of Biometric Recognition, 2008, (2)

[32] Upreti, S.R.: 'A new robust technique for optimal control of chemical engineering processes', Comput. Chem. Eng., 2004, 28, (8), pp. 1325-1336

[33] Mo, Y., Chen, D., and Hu, S.: 'Chaos particle swarm optimization algorithm and its application in biochemical process dynamic optimization', Huagong Xuebao/Journal of Chemical Industry and Engineering (China), 2006, 57, (9), pp. 2123-2127