

A Hybrid Estimation of Distribution Algorithm with Differential Evolution for Global Optimization

Bing Dong, Aimin Zhou and Guixu Zhang

Shanghai Key Laboratory of Multidimensional Information Processing

Department of Computer Science and Technology

East China Normal University, 3663 North Zhongshan Road, Shanghai 200062, China

Email: bing@stu.ecnu.edu.cn, {amzhou,gxzhang}@cs.ecnu.edu.cn

Abstract—In evolutionary algorithms, it is difficult to balance the exploration and exploitation. Usually, global search is utilized to find promising solutions, and local search is beneficial to the convergence of the solutions in the population. Combining different search strategies is a promising way to take advantages of different methods. Following the idea of DE/EDA, this paper proposes another way to combine estimation of distribution algorithm and differential evolution for global optimization. The basic idea is to choose either differential evolution or estimation of distribution algorithm for generating new trial solutions. To improve the algorithm performance, a local search strategy is used as well. The new approach, named as EDA/DE-EIG, is systematically compared with two state-of-art algorithms, and the experimental results show the advantages of our method.

Index Terms—DE, EDA, global optimization, eigenvector

I. INTRODUCTION

Estimation of distribution algorithms (EDAs) are stochastic optimization algorithms exploring the space of potential solutions [1]–[3]. Unlike traditional *evolutionary algorithms (EAs)*, there is no mutation or crossover in EDAs. They build explicit probabilistic models and sample from the built probabilistic models to obtain promising solutions. The explicit use of the probabilistic model has significant advantages over other methods, as it can utilize the global information of the population for producing more promising solutions.

Differential evolution (DE) has attracted a number of researchers from various background since proposed by Storn and Price [4], [5]. It uses the distance and direction information from the solutions in the current population to guide the process of search. DE is easy to be implemented and it has been widely applied to a variety of fields, i.e., control systems [6], robot control [7], remote sensing [8], electrical engineering [9], economic dispatch [10]. The main reasons of the popularity of DE are as follows [11]: (a) DE is much more simple to implement compared to other EAs with complex operations, (b) the parameters are fewer in comparison with other EAs (*CR*, *F*, *NP* in classical DE), and (c) recent researches have shown its superior performance on a wide variety of problems [12]–[14].

To improve the efficiency of DE for dealing with hard optimization problems, DE has also been hybridized with other search strategies, and the examples include DE/EDA [15], DE/BBO [16], DEPSO [17], DEFO [18], and etc.. Among them, DE/EDA utilizes the global information extracted by

EDA and the differential information exploited by DE to obtain promising solutions [15]. The differential information (i.e., direction and distance) from the solutions of the population is beneficial to accelerate the convergence speed, whereas the global information produced by EDA can guide the search to a more promising direction. By combining DE and EDA, ED/EDA is able to utilize both of the information and thus to balance the exploration and exploitation.

Following the idea of DE/EDA, this paper proposes another way to combine DE and EDA. The basic idea is to use either a DE operator or an EDA operator to generate a new trial solution. The new approach, named as EDA/DE-EIG, combines EDA with an enhanced DE [19], which utilizes an eigenvector-based crossover operator. Meanwhile, in order to improve the performance and avoid the solutions converged without obtaining optimal solutions, a local search is incorporated into this algorithm.

The remainder of the paper is organized as follows. In Section II, the definition of the problem is presented, and the basic algorithm frameworks of DE and EDA are introduced as well. Section III gives the details of the proposed algorithm EDA/DE-EIG. In Section IV, a systematic experiment is conducted to show the performance of EDA/DE-EIG. Finally, the paper is concluded in Section V.

II. RELATED WORK

This section firstly defines the global optimization problem. Then the basic algorithm frameworks of DE and EDA algorithms are introduced.

A. Definition

The box-constrained continuous global optimization can be stated in the following:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in [a_i, b_i]^n \end{aligned} \quad (1)$$

where $x = (x_1, x_2, \dots, x_n)^T \in R^n$ is a decision vector, $[a_i, b_i]^n$ is the search space and $f : R^n \rightarrow R$ is the objective function.

B. Differential Evolution

DE is a population-based global optimization algorithm. There are mainly three operations in classical DE, i.e., mutation, crossover and selection. The main algorithm framework of classic DE is given in Algorithm 1.

Algorithm 1: DE

```

1 Initial the population  $P_0$  randomly :
   $P_0 = \{x_{1,D}, x_{2,D}, x_{3,D}, \dots, x_{N,D}\}$ 
2 while not terminate do
  // mutation
3    $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G})$ 
  // crossover
4   if  $\text{rand}_j(0, 1) \leq CR$  or  $j = j_{\text{rand}}$  then
5      $u_{i,j,G} = v_{i,j,G}$ 
6   else
7      $u_{i,j,G} = x_{i,j,G}$ 
8   end
  // selection
9   if  $f(u_{i,G}) \leq f(x_{i,G})$  then
10     $x_{i,G+1} = u_{i,G}$ 
11  else
12     $x_{i,G+1} = x_{i,G}$ 
13  end
14 end

```

Some details of DE is given as follows.

- Initial population: The points in the population are the target vectors. N is the size of the population, and D is the dimension of the target vector.
- Mutation: The rand-1-bin mutation schema is used here, and there are other schemas that can be utilized in DE. At each generation G , a mutant vector $v_{i,G}$ is obtained by the mutation operator. F is the scaling factor, $r1, r2, r3$ are mutually different integers randomly selected from $[1, N]$ and also different from i .
- Crossover: The trial vector $u_{i,j,G}$ is generated by combing the mutant vector $v_{i,G}$ and the target vector $x_{i,j,G}$ according to the crossover operator. Hereby, $\text{rand}_j(0, 1)$ is a uniformly distributed random number between 0 and 1, and j_{rand} is a random integer between j and D , avoiding the trial vector is totally same as the target vector. CR is the controlling parameter.
- Selection: The trial vector $u_{i,G}$ and the target vector $x_{i,G}$ compete to enter the next generation in accordance with the objective function value.

C. Estimation of Distribution Algorithm

EDA is an emerging algorithm for the optimization, and it is distinct from traditional EAs. EDA consists of three main steps, including modeling, sampling and selection generally. The basic framework of EDA is shown in Algorithm 2.

III. OUR METHOD

DE/EDA is a hybrid algorithm for global optimization based on the combination of DE and EDA. However, as the DE

Algorithm 2: EDA

```

1 Initialization: Initial the  $Pop(t)$  randomly, and  $t$  is the
  generation.
2 while not terminate do
3   Modeling: Build a probabilistic model  $p(x)$ 
    according to the statistical information of the  $Pop(t)$ .
4   Sampling: Generate a new solution set  $Q$  by
    sampling from the built probabilistic model  $p(x)$ .
5   Selection: Select from  $Q \cup Pop(t)$  to construct the
    next population  $Pop(t+1)$ . The selection criterion is
    the objective function value.
6    $t = t + 1$ 
7 end

```

utilized in DE is quite simple. A more promising DE algorithm DE-EIG is imported to improve the performance. DE/EDA generates the offspring generation by utilizing the DE and EDA by the *CRP*. Meanwhile, for the further improvement, expensive LS [20] is applied. This section will introduce the framework of DE/EDA, DE-EIG and expensive LS respectively. And the algorithm framework of EDA/DE-EIG is presented finally.

A. DE-EIG

As traditional DE operates the crossover in the original coordinate, it is inevitable to lose some statistical information. To utilize the statistical information of the population, the eigenvectors of the solutions have been applied to the crossover in DE [19], [21]. DE-EIG makes the crossover rotationally invariant by transform the coordinate system of the solutions in the population [19]. It has shown impressive advantages over BBOB 2012 [22] and CEC 2013 [23] benchmark functions and two real-world optimization problems in CEC 2011 [24] compared other six algorithms.

In DE-EIG, an alternate coordinate system is employed by the individuals during the crossover. The eigenvector of the population is utilized to transform the coordinate system. Meanwhile, to exploit the diversity of the population and prevent the premature of the solutions in the population, the candidate solutions are generated with the original coordinate system or the rotated coordinated system randomly by a parameter.

The Algorithm 3 is the main framework of DE-EIG.

B. DE/EDA

The EDA/DE-EIG is proposed on the basis of the framework of DE/EDA [15]. EDA is incorporated with DE to generate more promising solutions. Hence, the efficiency of the search is improved meanwhile. The main algorithm framework of DE/EDA is presented in Algorithm 4.

C. Expensive local search

It is noteworthy that EAs are not very good at refining promising solutions especially in the later stage. Hence, it is significant to apply other search methods to accelerate

Algorithm 3: DE-EIG

```
1 Initial the population  $Pop(t) = \{x_1, x_2, x_3, \dots, x_N\}$  ( $N$ 
  is the size of the population)
2 while not terminate do
3    $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G})$ 
4   if  $rand() < p$  then
5     if  $rand() \leq CR$  then
6        $u_{i,G} = v_{i,G}$ 
7     else
8        $u_{i,G} = x_{i,G}$ 
9     end
10  else
11    Compute the the eigenvector matrix  $E$  of  $x_{i,G}$ ,
    let  $E'$  be the inverse matrix.
12     $x'_{i,G} = E' \cdot x_{i,G}$ 
13     $v'_{i,G} = E' \cdot v_{i,G}$ 
14    if  $rand() \leq CR$  then
15       $u'_{i,G} = v'_{i,G}$ 
16    else
17       $u'_{i,G} = x'_{i,G}$ 
18    end
19     $u_{i,G} = E \cdot u'_{i,G}$ 
20  end
21  if  $f(u_{i,G}) \leq f(x_{i,G})$  then
22     $x_{i,G+1} = u_{i,G}$ 
23  else
24     $x_{i,G+1} = x_{i,G}$ 
25  end
26   $t = t + 1$ 
27 end
```

the convergence speed. For this purpose, the expensive local search (LS) is introduced to improve this condition [20]. For simplicity, the details of the algorithm will not be presented here.

D. EDA/DE-EIG

In order to combine an enhanced DE with EDA and balance the global search and local search, the EDA/DE-EIG is proposed. EDA is utilized to extract the statistical information of the population, and the DE-EIG is beneficial to obtain the most important information of the population. The CRP to allocate the resource of EDA and DE-EIG is crucial to the performance of the algorithm. It is relatively problem-dependent. Hence it is a significant topic to study the CRP setting of the algorithm.

In Algorithm 5, the CRP is utilized to generate solutions by DE-EIG and EDA. Both the two algorithms play an important role to generate promising solutions. For $Converge(\theta, G, G_e)$ at line 18, it is essential to judge whether to operate expensive local search. Full description can be obtained from [20].

IV. EXPERIMENTAL STUDY

In this section, EDA/DE-EIG will be compared with JADE [25] and DE/EDA [15]. The source codes of JADE are

Algorithm 4: DE/EDA

```
1 Generate population  $Pop(t)$  randomly consists of  $N$ 
  solutions  $x_1, x_2, \dots, x_N$  from the feasible search space.
while not terminate do
2   Construct the probabilistic model:
3    $p_k(x) = \prod_{i=1}^n \mathcal{N}(x_i; \mu_i, \sigma_i)$ 
4   For all  $j = 1, 2, \dots, n$ , produce a trial solution
      $u = (u_1, u_2, \dots, u_n)$ 
5   if  $rand() < CRP$  then
6      $u_j = \frac{(x_i)_j + (x_d)_j}{2} + F \cdot [(x_d)_j - (x_i)_j + (x_b)_j - (x_c)_j]$ 
7   else
8      $u_j$  is sampled according to  $\mathcal{N}(x_i; \mu_i, \sigma_i)$ 
9   end
10  where  $CRP$  is the controlling parameter.
11  if  $f(u) < f(x_i)$  then
12     $x_i^{t+1} = u$ 
13  else
14     $x_i^{t+1} = x_i^t$ 
15  end
16   $t = t + 1$ 
17 end
```

Algorithm 5: EDA/DE-EIG

```
1 Initial the population  $Pop(t) = \{x_1, x_2, x_3, \dots, x_N\}$  ( $N$ 
  is the size of the population)
2 while not terminate do
3   Construct the probabilistic model:
4    $p(x) = \prod_{i=1}^n \mathcal{N}(x_i; \mu_i, \sigma_i)$ 
5   Generate a trial solution  $u_{i,G}$  as follows:
6   if  $rand() < CRP$  then
7      $u_{i,G}$  is produced by DE-EIG.
8   else
9      $u_{i,G}$  is sampled from the probabilistic model
       $p(x)$ .
10  end
11  if  $f(u_{i,G}) < f(x_{i,G})$  then
12     $x_{i,G+1} = u_{i,G}$ 
13  else
14     $x_{i,G+1} = x_{i,G}$ 
15  end
16  if  $Converge(\theta, G, G_e)$  then
17    Operate the expensive local search.
18  end
19   $t = t + 1$ 
20 end
```

from the authors. DE/EDA is implemented by ourself. The test instances and parameter settings are introduced in this section. A comprehensive study of the experimental results will be presented to illustrate the impressive advantages of EDA/DE-EIG.

TABLE I
STATISTICAL RESULTS (*mean* \pm *std*) FOR THE THREE ALGORITHMS ON INSTANCES $f1 - f13$.

instances	EDA/DE-EIG	JADE	DE/EDA
$f1$	1.54e-159 \pm 5.11e-159	$3.90e - 127 \pm 2.74e - 126(+)$	$1.39e - 59 \pm 2.58e - 59(+)$
$f2$	1.02e-75 \pm 7.46e-76	$2.60e - 35 \pm 1.64e - 34(+)$	$5.15e - 28 \pm 4.68e - 28(+)$
$f3$	4.01e-35 \pm 8.47e-35	$7.79e - 35 \pm 2.51e - 34(\sim)$	$1.23e - 12 \pm 1.20e - 12(+)$
$f4$	5.01e-20 \pm 3.06e-19	$3.15e - 14 \pm 6.42e - 14(+)$	$9.90e - 12 \pm 2.69e - 11(+)$
$f5$	$1.46e - 29 \pm 2.62e - 29$	3.85e-30 \pm 9.58e-30(-)	$3.37e - 21 \pm 8.66e - 21(+)$
$f6$	0.00e+00 \pm 0.00e+00	0.00e+00 \pm 0.00e+00(\sim)	0.00e+00 \pm 0.00e+00(\sim)
$f7$	$3.60e - 03 \pm 1.00e - 03$	6.01e-04 \pm 2.23e-04(-)	$2.20e - 03 \pm 5.59e - 04(-)$
$f8$	$2.79e + 03 \pm 5.02e + 02$	4.74e+00 \pm 2.34e+01(-)	$1.82e + 03 \pm 6.72e + 02(-)$
$f9$	$6.23e + 00 \pm 2.21e + 00$	0.00e+00 \pm 0.00e+00(-)	$1.54e + 02 \pm 1.96e + 01(+)$
$f10$	4.44e-15 \pm 0.00e+00	4.44e-15 \pm 0.00e+00(\sim)	4.44e-15 \pm 0.00e+00(\sim)
$f11$	0.00e+00 \pm 0.00e+00	$1.48e - 04 \pm 1.05e - 03(\sim)$	$2.96e - 04 \pm 1.46e - 03(\sim)$
$f12$	1.57e-32 \pm 5.53e-48	1.57e-32 \pm 5.53e-48(\sim)	1.57e-32 \pm 5.53e-48(\sim)
$f13$	1.35e-32 \pm 1.11e-47	1.35e-32 \pm 1.11e-47(\sim) 3(+) \sim 6(\sim)4(-)	1.35e-32 \pm 1.11e-47(\sim) 6(+) \sim 5(\sim)2(-)

¹ The bold ones mean the best.

A. Algorithms for Comparison

JADE [25] is an adaptive DE with a novel mutation strategy "DE/current-to- p best" with optional external archive and updating control parameters. JADE has been compared with several state-of-art algorithm and performs impressively. DE/EDA [15] is a hybrid algorithm incorporating EDA with DE, and it is the main framework of EDA/DE-EIG meanwhile. The three algorithms will be compared to EDA/DE-EIG on the same test instances.

B. Test Instances

All the algorithms will be compared on the first 13 test instances from YYL test instances [26]. The global minimum objective value is 0 for all test instances. And the test instances can be categorized into four kinds: $f1 - f5$ are unimodal functions. $f6$ is a step function. And $f7$ is a function with white noise. $f8 - f13$ are multimodal functions with many local optimal solutions. Hence, the test instances can be able to assess the performance of the algorithms from various aspects.

C. Parameter Settings

To compare the performance of the algorithms fairly, the parameter setting will be set according to the setting in the corresponding papers. All the algorithms are implemented by Matlab and executed in the same computer. The parameter settings are as follow:

- 1) The dimension of the population will be set to be 30 for all test instances. All algorithms are run independently 50 times and stopped after 450,000 function evaluations.

- 2) JADE: The parameters $N = 150, p = 0.05, c = 0.1, F = 0.5$ and $CR = 0.9$ are recommended in [25].
- 3) DE/EDA: The parameters are set as: $N = 150, F = 0.5$ and $CRP = 0.9$, which are considered according to the experimental results in [15].
- 4) EDA/DE-EIG: The CRP is 0.5; F is set to be 0.5; CR is set to be 0.6; the parameter p to control the probability to operate crossover two coordinate systems is 0.5; the convergence threshold $\theta = 0.1$; the size of the population N is 150. For the parameter setting of expensive local search, it is same as that in [20].

D. Experimental Results and Analysis

Table I illustrates the mean and the standard deviation of the results obtained by the four algorithms after 450,000 FEs over 50 independent runs. The Wilcoxon rank sum test at 0.05 significance is performed to compare the function value obtained by EDA/DE-EIG to another algorithm. And "+", "-", and " \sim " respectively denotes the function objective value of another algorithm is larger than, less than, and similar to that of EDA/DE-EIG at 0.05 significance level by a Wilcoxon rank sum test.

From Table I, the impressive performance of EDA/DE-EIG is distinct with comparison to the other two algorithms for the mean value of the final results. EDA/DE-EIG obtain the best results on 9 test instances except $f5, f7, f8$ and $f9$. For further comparison, EDA/DE-EIG will be compared with the other two algorithms respectively. For DE/EDA, EDA/DE-EIG has a substantial improvement of the performance. EDA/DE-EIG

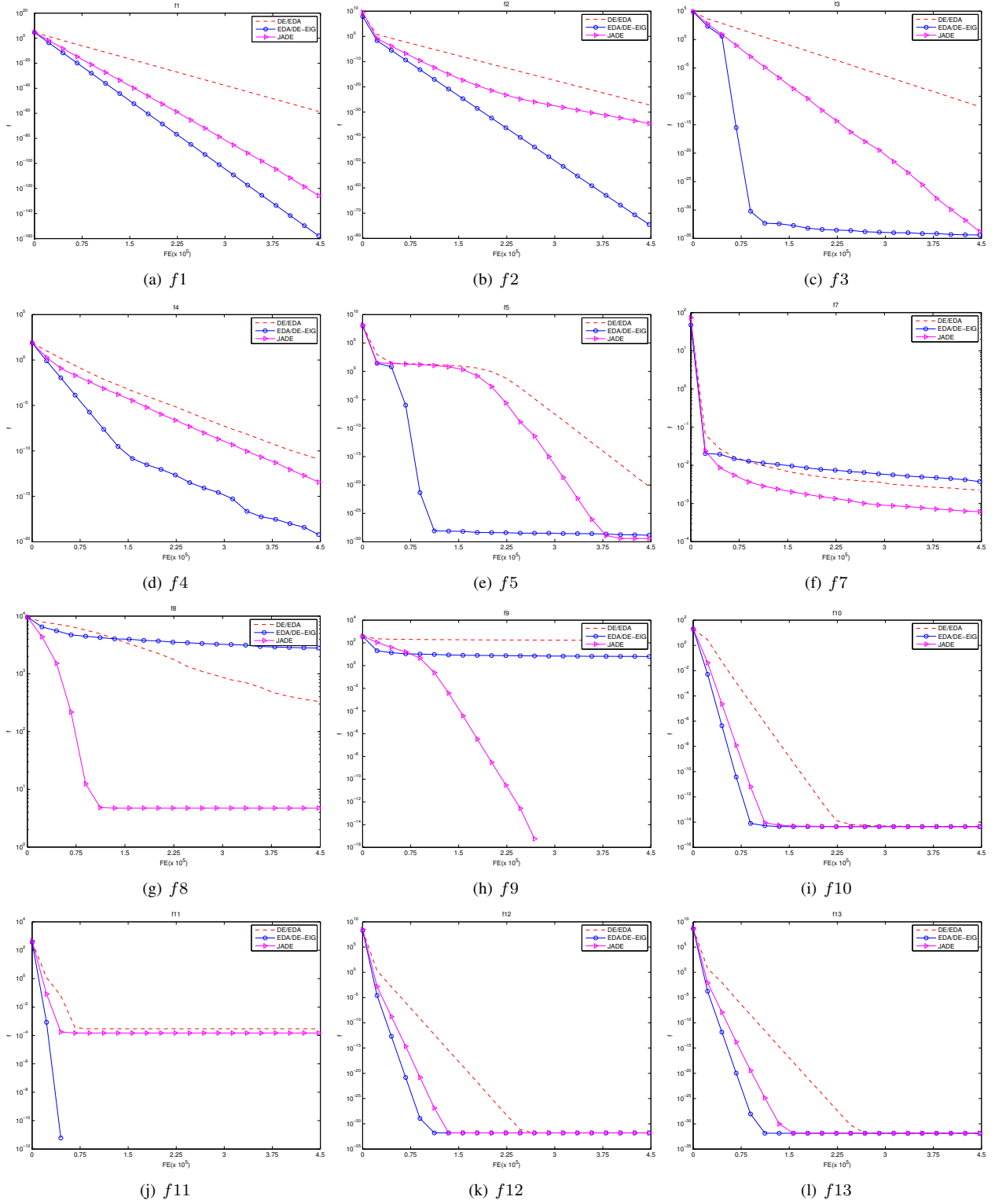


Fig. 1. The mean function value versus evaluation counts on the test instances.

performs better than DE/EDA on 7 test instances. EDA/DE-EIG has made a relatively significant improvement for the first five test instances and f_{11} . The difference between the two algorithms over these test instances are distinct.

Compare EDA/DE-EIG with JADE, EDA/DE-EIG has advantages over JADE on 5 test instances. And both the two algorithms obtain the best results on f_6 , f_{10} , f_{12} and f_{13} . In conclusion, the significant performance of EDA/DE-EIG is competitive. For $f_6, f_{10}, f_{12}, f_{13}$, the three algorithms have the same results. For f_{11} , only EDA/DE-EIG obtains the best result. JADE does not perform well enough because of worse performance of several runs. It should be noteworthy that EDA/DE-EIG has a distinct performance for $f_1 - f_4$ and f_{11} . As for f_7, f_8 and f_9 , EDA/DE-EIG may be impacted by the local optimum of DE. It is worthwhile to improve the performance of EDA/DE-EIG on these test instances.

Meanwhile, from a more objective prospect, the Wilcoxon rank sum test is implemented to compare the performance of EDA/DE-EIG with that of others. For JADE, EDA/DE-EIG outperforms on 3 test instances. And the two algorithms obtain the similar performance on 6 test instances. EDA/DE-EIG does not perform as well as JADE on 4 test instances. In general, EDA/DE-EIG has made a substantial improvement in comparison with DE/EDA. It does indicate the effect brought by DE-EIG. As for JADE, EDA/DE-EIG nearly has similar performance on 13 test instances statistically. Hence, the performance of EDA/DE-EIG is relatively promising and need further exploration.

Moreover, for a more detailed illustration of the performance of EDA/DE-EIG, the Figure 1 is the mean function value of the three algorithms, intuitively illustrates the comparison of the performance of the three algorithms on 12 test instances except f_6 . As the results of JADE and EDA/DE-EIG except DE/EDA converges very fast on f_6 . For a better presentation, the result of f_6 will not be presented here. Statistically, EDA/DE-EIG obtain the best results on 8 out of 12 test instances. EDA/DE-EIG is superior to the other two algorithms on convergence speed or the final solution. Especially for some instances, including f_1, f_2, f_3, f_4 and f_{11} , the improvement is extremely distinct. For DE/EDA, it performs worse than EDA/DE-EIG on 11 test instances except f_7 and f_8 . In comparison with JADE, EDA/DE-EIG outperforms on $f_1 - f_4$ both on convergence speed and the final solution. As for f_5 , though JADE obtains a better solution. EDA/DE-EIG has a more promising convergence trend and converge faster. For f_7, f_8 and f_9 , EDA/DE-EIG does not have a desired performance. Hence, the performance of ED-EIG to improve the solution need more exploration. For f_{10}, f_{12} and f_{13} , EDA/DE-EIG has advantages on the convergence speed. In conclusion, EDA/DE-EIG has made a impressive improvement of DE/EDA. Meanwhile, with comparison to JADE, EDA/DE-EIG has the advantages on convergence speed or the final solution. In conclusion, the performance of EDA/DE-EIG is relatively competitive.

V. CONCLUSIONS AND FUTURE WORK

DE/EDA [15] is a promising method that utilizes both the the global and local information for global optimization. However, the potential improvement of the performance of this algorithm has not been exploited furthermore. In this paper, an improved DE, DE-EIG, is imported to combine with EDA, bringing an impressive improvement on the performance. DE-EIG is beneficial to utilize the statistics information of the population to accelerate the convergence. And expensive local search is applied to improve the performance further. The experimental results have shown the distinct advantages of the proposed method, named as EDA/DE-EIG, in comparison with two state-of-art algorithms JADE [25] and DE/EDA [15].

The results reported in this paper is preliminary and there are several ways to improve the algorithm performance. Firstly, the algorithm framework of EDA/DE-EIG can be simplified. Secondly, it is worth to investigate how to allocate the computational resources to both DE-EIG and EDA.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China under Grant No.61273313 and 61673180, and the Science and Technology Commission of Shanghai Municipality under Grant No.14DZ2260800.

REFERENCES

- [1] S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning," DTIC Document, Tech. Rep., 1994.
- [2] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media, 2002, vol. 2.
- [3] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Computational optimization and applications*, vol. 21, no. 1, pp. 5–20, 2002.
- [4] R. Storn and K. Price, *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*. ICSI Berkeley, 1995, vol. 3.
- [5] —, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [6] A. Biswas, S. Das, A. Abraham, and S. Dasgupta, "Design of fractional-order $PI^\lambda D^\mu$ controllers with an improved differential evolution," *Engineering applications of artificial intelligence*, vol. 22, no. 2, pp. 343–350, 2009.
- [7] F. Neri and E. Mininno, "Memetic compact differential evolution for cartesian robot control," *Computational Intelligence Magazine, IEEE*, vol. 5, no. 2, pp. 54–65, 2010.
- [8] Y. Zhong, S. Zhang, and L. Zhang, "Automatic fuzzy clustering based on adaptive multi-objective differential evolution for remote sensing imagery," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 6, no. 5, pp. 2290–2301, 2013.
- [9] B. Mohanty, S. Panda, and P. Hota, "Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 77–85, 2014.
- [10] S. Sayah and A. Hamouda, "A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems," *Applied Soft Computing*, vol. 13, no. 4, pp. 1608–1619, 2013.
- [11] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, 2011.
- [12] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "Opposition-based differential evolution," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 1, pp. 64–79, 2008.

- [13] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 3, pp. 526–553, 2009.
- [14] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, 2011.
- [15] J. Sun, Q. Zhang, and E. P. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Information Sciences*, vol. 169, no. 3, pp. 249–262, 2005.
- [16] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2010.
- [17] W.-J. Zhang, X.-F. Xie *et al.*, "DEPSO: hybrid particle swarm with differential evolution operator," in *IEEE International Conference on Systems Man and Cybernetics*, vol. 4, 2003, pp. 3816–3821.
- [18] Y.-J. Zheng, X.-L. Xu, H.-F. Ling, and S.-Y. Chen, "A hybrid fireworks optimization method with differential evolution operators," *Neurocomputing*, vol. 148, pp. 75–82, 2015.
- [19] S.-M. Guo and C.-C. Yang, "Enhancing differential evolution utilizing eigenvector-based crossover operator," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 1, pp. 31–49, 2015.
- [20] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 6, pp. 807–822, 2015.
- [21] Y. Wang, H.-X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, pp. 232–247, 2014.
- [22] N. Hansen, A. Auger, S. Finck, and R. Ros, "real-parameter black-box optimization benchmarking 2012 :Experiment setup," INRIA, France, Tech. Rep., 2012.
- [23] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, 2013.
- [24] S. Das and P. Suganthan, "Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems," *Jadavpur University, Nanyang Technological University, Kolkata*, 2010.
- [25] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 945–958, 2009.
- [26] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, 1999.
- [27] M. Crepinsek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM COMPUTING SURVEYS*, vol. 45, no. 3, 2013.