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, East China Normal University
500 Dongchuan Road, Shanghai, China
Email: bing@stu.ecnu.edu.cn, {amzhou,gxzhang}@cs.ecnu.edu.cn

***Abstract*—In evolutionary algorithms (EAs), it is difficult to balance the global search and local exploitation. Global search is utilized to find promising solutions, and local exploitation is beneficial to the convergence of the solutions in the population. Addressing this issue, DE/EDA combing differential evolution (DE) with estimation of distribution algorithm (EDA) is proposed for global optimization. The control parameter (CRP) controlling the resource allocation of DE and EDA in DE/EDA plays an important role. We combine the algorithm with several suitable CRP settings to obtain trial solutions, and select the best ones into the next generation according to the function value. Moreover, an enhanced DE based on an eigenvector-based crossover operator is imported into this algorithm. Hence, the hybrid estimation of distribution algorithm with differential evolution (EDA/DE-EIG) is proposed for global optimization. The algorithm is compared to three state-of-art algorithms : JADE, EDA/LS and DE/EDA. The experimental results show the impressive advantages of EDA/DE-EIG.**

***Index Terms*—DE, EDA, global optimization, eigenvector**

I. Introduction

EDAs are stochastic optimization algorithms exploring the space of potential solutions [1]–[3]. Unlike traditional EAs, there is no mutation or crossover in EDAs. They build an explicit probabilistic model and sample from the built probabilistic model 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 to produce more promising solutions.

DE has attracted a number of researchers from various background since proposed by Storn and Price [4], [5]. DE is easy to be implemented, whereas it performs impressively for many problems. It utilizes the distance and direction information from the population to guide the process of search. DE has been widely applied to a variety of fields, i.e., control systems [6], robot control [7], remote sensing [8], electrical engineering [9]. The main reasons of the popularity of DE are as follows [10]: (1) DE is much more simple to be implemented compared to other evolutionary algorithms with complex operations. (2) The parameters are fewer in comparison with other EAs (CR, F, NP in classical DE). (3) Meanwhile, recent researches about DE have shown its superior performance on a wide variety of problems [11]–[13].

Due to the simplicity and powerful performance of DE, it has been combined with other EAs to improve the performance, i.e., DE/EDA [14], DE/BBO [15], DEPSO [16]. DE/EDA is one of the hybrid EAs, and it utilizes the global information extracted from EDA and the differential information exploited by DE to obtain promising solutions [14]. DE obtains differential information (i.e., direction and distance) from the solutions of the population. It is beneficial to the further search and accelerate the convergence speed. However, DE is unable to utilize global information related to the search space. EDA extracts statistical global information directly by sampling from the built probabilistic model. Hence, the DE/EDA is proposed to make use of statistical global information and local information of the solutions. The local information obtained by DE can accelerate the exploitation, whereas the global information produced by EDA can guide the search to a more promising direction. Based on the framework of DE/EDA, more further work can be researched by exploring the ability of more powerful DE algorithms.

However, there is still a troublesome problem of the setting of the CRP, controlling the resource allocation of DE and EDA in DE/EDA. If the CRP is too large, it will be hard to find a optimal solution. And the solutions will be hard to converge if the CRP is small. Hence, it is apparent that a fixed setting of CRP is unable to solve various test instances. It is significant to balance the resource allocation of DE and EDA with a more appropriate strategy.

Based on the above considerations, EDA is combined with an enhanced DE [17], which utilizes an eigenvector-based crossover operator. Inspired by [13], we combine DE/EDA with several common CRP settings to produce several trial solutions, and the best solutions are selected into the next generation in terms of the function objective value. Meanwhile, in order to improve the performance and avoid the solutions converged without obtaining optimal solutions, the expensive local search (LS) is incorporated into this algorithm. Thus, EDA/DE-EIG is proposed for global optimization.

The remainder of the paper is organized as follows. In section II, the definition of the problem will be presented. And the basic algorithm framework of DE and EDA will be introduced. Section III illustrates the related algorithms and the proposed algorithm EDA/DE-EIG in detail. In section IV, a systematic experimental results will prove the advantages of EDA/DE-EIG. Finally, the conclusion and the prospect will be presented in section V.

II. Defination and Algorithms

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

A. Definition

The continuous global optimization will be stated in the following:

$$\begin{aligned} & minf(x) \\ & s.t. x \in [a_i, b_i]^n \end{aligned} \tag{1}$$

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

B. DE

DE is a population-based global optimization algorithm. There are mainly three operations in classical DE: mutation, crossover and selection. The main algorithm framework of classic DE is as follows:

Algorithm 1: DE

Initial the population P_0 randomly : $P_0 = \{x_{1,D}, x_{2,D}, x_{3,D}, \cdots, x_{N,D}\}$
while *not terminate* **do**
  // mutation
  $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G})$
  // crossover
  if $rand_j(0,1) \leq CR$ *or* $j = j_{rand}$ **then**
    $u_{i,j,G} = v_{i,j,G}$
  else
    $u_{i,j,G} = x_{i,j,G}$
  end
  // selection
  if $f(u_{i,G}) \leq f(x_{i,G})$ **then**
    $x_{i,G+1} = u_{i,G}$
  else
    $x_{i,G+1} = x_{i,G}$
  end
end

- Initial population: The points in the population are the target vector. N is the size of the population, and D is the dimension of the target vector.
- Mutation: 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, $rand_j(0,1)$ 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. EDA

EDA is an emerging algorithm for the optimization, and it is distinct from traditional EAs. EDA consists of three main steps, including modelling, sampling and selection generally.

Algorithm 2: EDA

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

III. EDA/DE-EIG

DE/EDA is a hybrid algorithm for global optimization based on the combination of DE and EDA. However, a simple combination of DE and EDA cannot satisfy more complex problems. 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. However, how to allocate the resource to DE and EDA to generate offspring generation is not a trivial task. A fixed set of CRP is unable to face the challenges from more complex problems. Hence, a CRP parameter pool is adopted to generate trial solutions, and the best one is selected according to the function value. This compositional parameter setting schema is described in the Figure 1. Meanwhile, for the further improvement, expensive LS [18] 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 [17], [19]. DE-EIG makes the crossover rotationally invariant by transform the coordinate system of the solutions in the population [17]. It has shown impressive advantages over BBOB 2012 [20] and CEC 2013 [21] benchmark functions and two real-world optimization problems in CEC 2011 [22] compared other six algorithms.

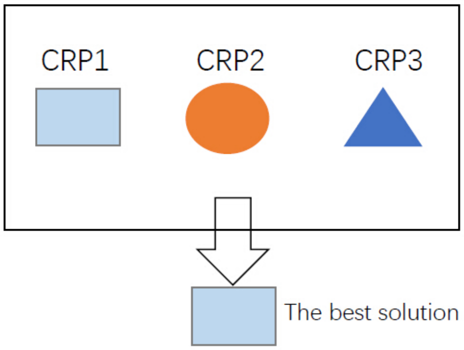


Fig. 1. Adopt the CRP pool to generate the best solution

Algorithm 4: DE/EDA

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

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 frame work of DE/EDA [14]. 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.

Algorithm 3: DE-EIG

Initial the population $Pop(t) = \{x_1, x_2, x_3, \cdots, x_N\}$ (N is the size of the population)
while *not terminate* **do**
  $v_{i,G} = x_{i,G}(r_1, :) + F \cdot (x_{i,G}(r_2, :) - x_{i,G}(r_3, :))$
  if $rand() < p$ **then**
    if $rand() \leq CR$ **then**
      $u_{i,G} = v_{i,G}$
    else
      $u_{i,G} = x_{i,G}$
    end
  else
    Compute the the eigenvector matrix E of $x_{i,G}$, and E' is the inverse matrix.
    $x'_{i,G} = E' \cdot x_{i,G}$
    $v'_{i,G} = E' \cdot v_{i,G}$
    if $rand() \leq CR$ **then**
      $u'_{i,G} = v'_{i,G}$
    else
      $u'_{i,G} = x'_{i,G}$
    end
    $u_{i,G} = E \cdot u'_{i,G}$
  end
  if $f(u_{i,G}) \leq f(x_{i,G})$ **then**
    $x_{i,G+1} = u_{i,G}$
  else
    $x_{i,G+1} = x_{i,G}$
  end
  $t = t + 1$
end

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 the convergence speed. For this purpose, the expensive local search (LS) is introduced to improve this condition [18]. 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. However, it is not a trivial task to allocate appropriate resource to DE and EDA respectively. It is noteworthy that a fixed allocation of resource to DE and EDA is unable to solve various problems. Hence, to improve the robustness of the algorithm, EDA/DE-EIG is combined with several proper CRP settings. With this strategy, more trial solutions are generated and the best ones are selected into the next generation. The following is the algorithm framework of EDA/DE-EIG.

Algorithm 5: EDA/DE-EIG

Initial the population $Pop(t) = \{x_1, x_2, x_3, \cdots, x_N\}$ (N is the size of the population)
while *not terminate* **do**
  Construct the probabilistic model:
  $p(x) = \prod_{i=1}^{n} \mathcal{N}(x_i; \mu_i, \sigma_i)$
  Set the parameter pool $CRP = [c_1, c_2, c_3]$
  Generate a trial solution $u = (u_1, u_2, \cdots, u_n)$ as follows:
  if $rand() < CRP(j)$ *or* $j = j_{rand}$ **then**
    $u_{i,j}$ is produced by DE-EIG.
  else
    $u_{i,j}$ is sampled from the probabilistic model $p(x)$.
  end
  $u_{i,G}$ is selected from $u_{i,j} (j = 1, 2, 3)$
  if $f(u_{i,G}) < f(x_{i,G})$ **then**
    $x_{i,G+1} = u_{i,G}$
  else
    $x_{i,G+1} = x_{i,G}$
  end
  if *Converage*(θ, G, G_e) **then**
    Operate the expensive local search.
  end
  $t = t + 1$
end

IV. Experimental study

In this section, EDA/DE-EIG will be compared with three algorithms : JADE, EDA/LS and DE/EDA. The source codes of JADE and EDA/LS are 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.

A. Algorithms for comparison

JADE [23] is an adaptive DE with a novel mutation s-trategy "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. EDA/LS [18] is a hybrid algorithm combining EDA with cheap and expensive local search methods. In this method, part of the candidate solutions are sampled from a modified probabilistic model and the rest are obtained by refining the parent solutions through a cheap LS. And when the population converged, an expensive LS will be applied to improve the quality of the solution found so far. DE/EDA [14] 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 [24]. 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 3000 generations.
2) EDA/LS: The size of the population N is 150; the number of the bins used in the probabilistic model is $M = 15$; the percentage of best solutions P_b used in the local surrogate model is 0.2; the probability P_c to use location information is 0.2; and the convergence threshold $\theta = 0.1$.
3) JADE: The parameters $N = 100, p = 0.05, c = 0.1, F = 0.5$ and $CR = 0.9$ are recommended in [23].
4) DE/EDA: The parameters are set as: $N = 150, F = 0.5$ and $\theta = 0.9$, which are considered according to the experimental results in [14].
5) EDA/DE-EIG: The CRP parameter pool is [0.1, 0.5, 0.9]; 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.

D. Study of the results

Table I illustrates the mean and the standard deviation of the results obtained by the four algorithms after 3000 generations 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 "∼" 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. Table II summarizes the average generations (AG) and number of success runs (SR) to achieve the value to reach ($VTR = 1.0 \times -14$). AR [25] is the criteria to assess the comparison of the convergence speed between EDA/DE-EIG and other algorithms, where $AR > 1$ indicates EDA/DE-EIG converges faster than other algorithms. AR is intuitive to illustrates the comparison of the convergence speed.

From Table I, the impressive performance of EDA/DE-EIG is distinct with comparison to the other three algorithms. EDA/DE-EIG obtain the best results on 12 test instances except $f9$. For further comparison, EDA/DE-EIG will be compared with the other three algorithms respectively. For DE/EDA, EDA/DE-EIG has a substantial improvement of the performance. EDA/DE-EIG performs better than DE/EDA on

TABLE I

STATISTICAL RESULTS ($mean \pm std$) FOR THE FOUR ALGORITHMS ON INSTANCES $f1 - f13$.

instances	EDA/DE-EIG	JADE	EDA/LS	DE/EDA
$f1$	**2.29e-252** $\pm$ **0.00e+00**	$3.90e-127 \pm 2.74e-126(+)$	$2.59e-190 \pm 0.00e+00(+)$	$1.39e-59 \pm 2.58e-59(+)$
$f2$	**4.39e-120** $\pm$ **2.64e-120**	$2.60e-35 \pm 1.64e-34(+)$	$6.79e-98 \pm 4.54e-98(+)$	$5.15e-28 \pm 4.68e-28(+)$
$f3$	**1.43e-41** $\pm$ **1.01e-40**	$7.79e-35 \pm 2.51e-34(+)$	$1.36e-35 \pm 8.00e-35(+)$	$1.23e-12 \pm 1.20e-12(+)$
$f4$	**8.30e-59** $\pm$ **4.94e-59**	$3.15e-14 \pm 6.42e-14(+)$	$3.31e-55 \pm 1.12e-54(+)$	$9.90e-12 \pm 2.69e-11(+)$
$f5$	**0.00e+00** $\pm$ **0.00e+00**	$3.85e-30 \pm 9.58e-30(+)$	$5.98e-30 \pm 1.18e-29(+)$	$3.37e-21 \pm 8.66e-21(+)$
$f6$	**0.00e+00** $\pm$ **0.00e+00**	**0.00e+00** $\pm$ **0.00e+00**$(\sim)$	**0.00e+00** $\pm$ **0.00e+00**$(\sim)$	**0.00e+00** $\pm$ **0.00e+00**$(\sim)$
$f7$	**4.31e-04** $\pm$ **1.01e-04**	$6.01e-04 \pm 2.23e-04(+)$	$1.80e-03 \pm 5.08e-04(+)$	$2.20e-03 \pm 5.59e-04(+)$
$f8$	**0.00e+00** $\pm$ **0.00e+00**	$4.74e+00 \pm 2.34e+01(\sim)$	**0.00e+00** $\pm$ **0.00e+00**$(\sim)$	$1.82e+03 \pm 6.72e+02(+)$
$f9$	$6.55e+00 \pm 2.63e+00$	**0.00e+00** $\pm$ **0.00e+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**$(\sim)$	**4.44e-15** $\pm$ **0.00e+00**$(\sim)$
$f11$	**0.00e+00** $\pm$ **0.00e+00**	$1.48e-04 \pm 1.05e-03(\sim)$	**0.00e+00** $\pm$ **0.00e+00**$(\sim)$	$2.96e-04 \pm 1.46e-03(\sim)$
$f12$	**1.57e-32** $\pm$ **5.53e-48**	**1.57e-32** $\pm$ **5.53e-48**$(\sim)$	**1.57e-32** $\pm$ **5.53e-48**$(\sim)$	**1.57e-32** $\pm$ **5.53e-48**$(\sim)$
$f13$	**1.36e-32** $\pm$ **1.11e-47**	**1.36e-32** $\pm$ **1.11e-47**$(\sim)$	**1.35e-32** $\pm$ **1.11e-47**$(\sim)$	**1.35e-32** $\pm$ **1.11e-47**$(\sim)$
		$6(+)6(\sim)1(-)$	$6(+)6(\sim)1(-)$	$8(+)5(\sim)$

[1] The bolder ones mean the best.

TABLE II

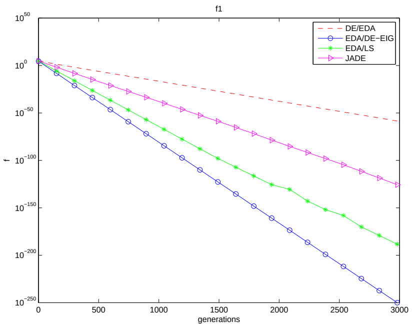
STATISTICAL RESULTS OF AVERAGE GENERATIONS (AG) AND THE NUMBER OF SUCCESS RUNS (SR) TO ACHIEVE $VTR = 1.0 \times 10^{-14}$ ON $f1$-$f13$ AFTER 3000 GENERATIONS OVER 50 RUNS FOR EDA/DE-EIG, JADE, EDA/LS AND DE/EDA.

instances	EDA/DE-EIG		JADE		EDA/LS		DE/EDA		AR		
	AG	SR	AG	SR	AG	SR	AG	SR	1. vs 2.	1. vs 3.	1. vs 4.
$f1$	**2.17e+02**	50	$4.26e+02$	50	$2.70e+02$	50	$8.73e+02$	50	1.96	1.24	4.02
$f2$	**3.99e+02**	50	$8.09e+02$	50	$4.84e+02$	50	$1.64e+03$	50	2.03	1.21	4.11
$f3$	**6.26e+02**	50	$1.40e+03$	50	$1.03e+03$	50	NA	0	2.24	1.65	NA
$f4$	$7.92e+02$	50	$1.70e+03$	31	**7.29e+02**	50	$2.91e+03$	14	2.15	0.92	3.67
$f5$	$5.70e+02$	50	$1.75e+03$	50	**5.68e+02**	50	$2.45e+03$	50	3.07	0.99	4.30
$f6$	**5.45e+01**	50	$1.09e+02$	50	$6.40e+01$	50	$2.22e+02$	50	2.00	1.17	4.07
$f7$	NA	0	NA	0	NA	0	NA	0	NA	NA	NA
$f8$	$1.19e+03$	50	$1.59e+03$	48	**4.63e+02**	50	$1.45e+02$	4	1.33	0.39	1.52
$f9$	NA	0	$1.73e+03$	50	**1.20e+03**	50	NA	0	NA	NA	NA
$f10$	**3.73e+02**	50	$7.23e+02$	50	$4.62e+02$	50	$1.50e+03$	50	1.94	1.23	4.02
$f11$	**2.23e+02**	50	$4.96e+02$	49	$2.78e+02$	50	$8.96e+02$	48	2.22	1.25	4.02
$f12$	**2.05e+02**	50	$4.17e+02$	50	$2.46e+02$	50	$8.19e+02$	50	2.03	1.20	4.00
$f13$	**2.16e+02**	50	$4.42e+02$	50	$2.58e+02$	50	$8.56e+02$	50	2.05	1.19	3.96

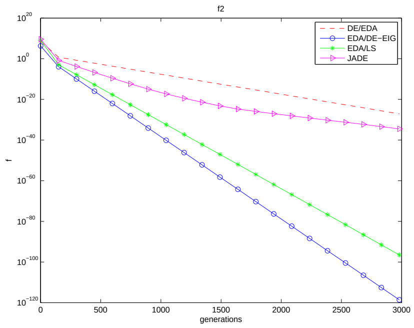
[1] $AR = AG_{other}/AG_{EDA/DE-EIG}$ is employed to compare the convergence speed between EDA/DE-EIG and other algorithms.
[2] "1. vs. 2." means "EDA/DE-EIG vs. JADE", "1. vs. 3." means "EDA/DE-EIG vs. EDA/LS", and "1. vs. 4." means "EDA/DE-EIG vs. DE/EDA".
[3] The bolder one means the best.

9 test instances. Meanwhile, the differences of the mean value and standard deviation value of the two algorithms are distinct, especially for $f1 - f5$ and $f8$. Compare EDA/DE-EIG with JADE, EDA/DE-EIG has advantages over JADE on 8 test instances. In comparison EDA/LS with EDA/DE-EIG, the two algorithms have the same performance on 6 test instances. EDA/DE-EIG outperforms than EDA/LS on 6 test instances. In conclusion, the significant performance of EDA/DE-EIG is evident. For $f6, f10, f12, f13$, the four algorithms have the same results. For $f11$, both EDA/DE-EIG and EDA/LS have the same performance. As for other test instances except $f9$, EDA/DE-EIG obtain the best results. And the performance of EDA/DE-EIG on $f1 - f5$ and $f7$ is unreachable to the other three algorithms. Except for $f9$, it may be impacted by DE for local optimum.

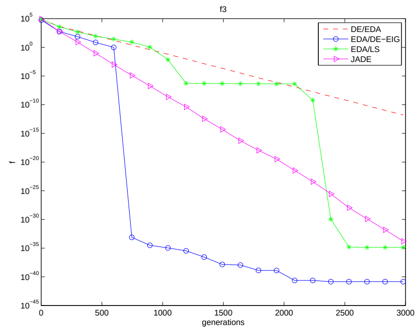
Meanwhile, from a more objective prospect, the Wilcoxon rank sum test is implemented to compare the performance of EDA/DE-EIG with that of others. For EDA/LS and JADE, EDA/DE-EIG obtain the better results on 6 test instances, and performs similar to the two algorithms on 6 test in-



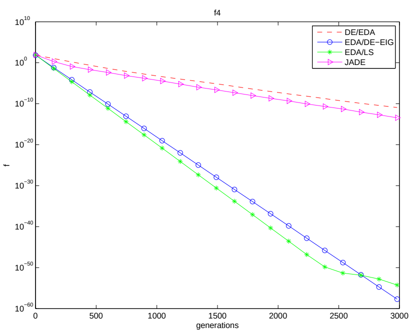
(a) $f1$



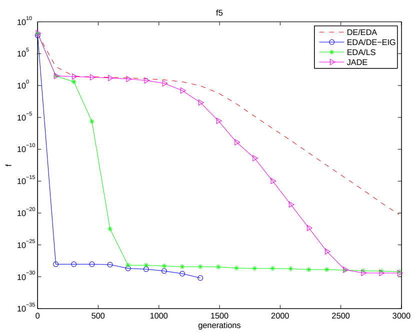
(b) $f2$



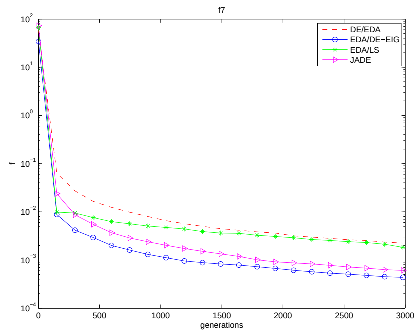
(c) $f3$



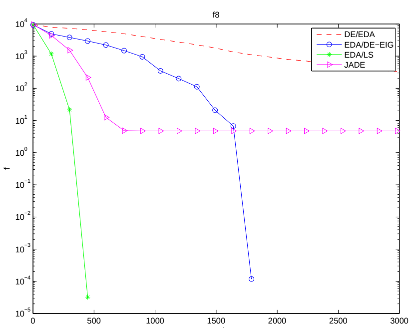
(d) $f4$



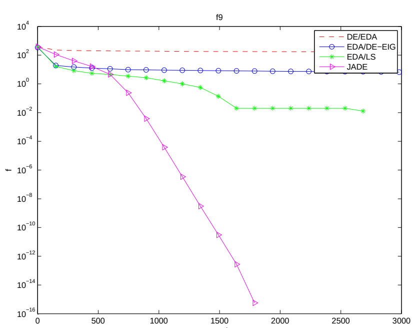
(e) $f5$



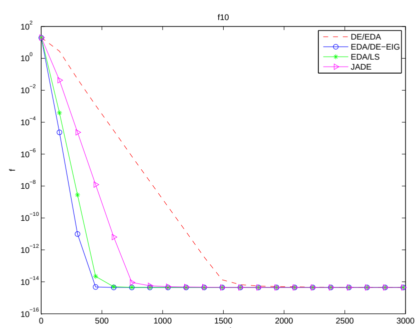
(f) $f7$



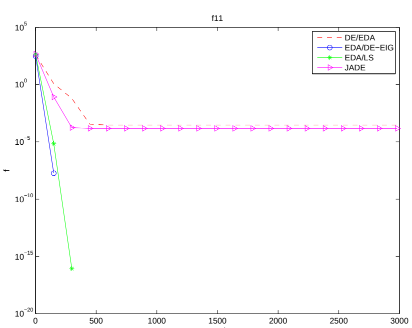
(g) $f8$



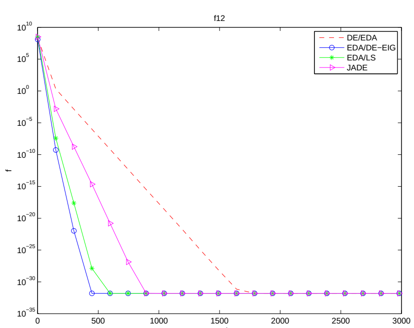
(h) $f9$



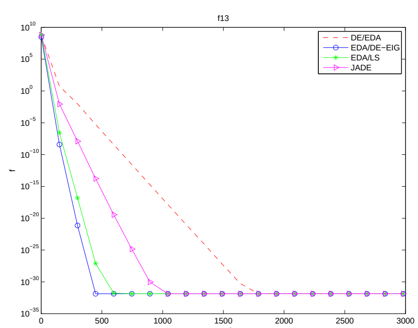
(i) $f10$



(j) $f11$



(k) $f12$



(l) $f13$

Fig. 2. The mean function value of $f1 - f13$ except $f6$ of the four algorithms.

stances. Only for $f9$, EDA/LS and JADE has advantages over EDA/DE-EIG. As for DE/EDA, EDA/DE-EIG has a substantial improvement on 8 test instances. On the other 5 test instances, the performance of the two algorithms are similar. In general, EDA/DE-EIG is substantially superior to the other three algorithms.

The Table II is evident to illustrate the comparison of the convergence speed of the four algorithms. AG is a powerful evidence to manifest the convergence speed. EDA/DE-EIG converges fastest on 8 test instances. SR is utilized to study the steady of the performance of the algorithms. EDA/DE-EIG run successfully on 11 test instances except $f7$ and $f9$. SR of EDA/LS is 50 on 12 test instances. For JADE, SR is not 50 on 4 test instances. DE/DEA is the algorithm performs worst, SR of this algorithm fails to reach 50 on 6 test instances. AR is the supplementary evidence to demonstrate the comparison of the convergence speed of EDA/DE-EIG and the three other algorithms. And the convergence speed of EDA/DE-EIG is superior to the other three algorithms impressively. Especially for DE/EDA, AR exceeds 3 for the most test instances. The comparison of the convergence speed between EDA/DE-EIG and DE/EDA is distinct. On $f4$ and $f5$, EDA/LS converges nearly to EDA/DE-EIG. However, on 8 test instances, the convergence speed of EDA/DE-EIG is more satisfying. As for JADE, EDA/DE-EIG is superior to JADE on most test instances except $f9$. Overall, from Table II, the final results of the four algorithms on $f6$, $f11$, $f12$ and $f13$ are similar. Comprehensively, the convergence speed of EDA/DE-EIG is impressive with comparison with the other three algorithms. However, the distinct advantages of the convergence speed of EDA/DE-EIG is proved by the comparison of the AR in Table II. Hence, EDA/DE-EIG has an impressive performance both on the final results and convergence speed.

Moreover, for a more detailed illustration of the performance of EDA/DE-EIG, the Figure 2 is the mean function value of the four algorithms via generations, intuitively illustrates the comparison of the performance of the four algorithms on 12 test instances except $f6$. As the results of three algorithms except DE/EDA converges very fast on $f6$. For a better presentation, the result of $f6$ will not be presented here. Statistically, EDA/DE-EIG obtain the best results on 9 out of 12 test instances. EDA/DE-EIG is superior to the other three algorithms both on convergence speed and the final solution. Especially for some instances, including $f1$, $f2$, $f3$ and $f5$, the improvement is extremely distinct. For DE/EDA,it performs worse than EDA/DE-EIG on any test instance. Substantially, the DE-EIG and the strategy of the allocation of the resources are significant to the improvement of the performance. Though EDA/LS converges faster than EDA/DE-EIG in the font half past generations. EDA/DE-EIG converges to 0 in the nearly 1800 generations which is not reachable to EDA/LS. JADE outperforms than EDA/DE-EIG on only one test instance $f9$. On $f8$, though JADE converges fast than EDA/DE-EIG in the early generations. EDA/DE-EIG obtains a better final solution than JADE. EDA/LS is a recently proposed algorithm with impressive performance. EDA/DE-EIG outperforms than EDA/LS on 9 test instances. For $f1$, $f2$, $f3$ and $f5$, EDA/DE-EIG has a substantial improvement on the convergence speed and the final result. On $f4$, EDA/LS converges slightly fast than EDA/DE in the most generations. But the final solution of EDA/LS is larger than that of EDA/DE-EIG. In conclusion, except for $f9$, the performance of EDA/DE-EIG is impressively competitive.

V. CONCLUSION AND FUTURE WORK

To balance the global search and local information in EAs, DE/EDA is a promising method addressing this problem. 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. Moreover, in order to allocate the resource to DE and EDA reasonably, a CRP parameter pool is utilized to generate trial solutions and select the best solution for the next generation. This strategy is significant to generate the most promising trial solutions. And expensive LS is applied to improve the performance further. The experimental results have shown the distinct advantages of EDA/DE-EIG in comparison with three state-of-art algorithms: JADE, EDA/LS and DE/EDA. It is significant to combine EDA with powerful DE algorithms, as it will be efficient to utilize the global information and local information.

The algorithm framework of EDA/DE-EIG need to be simplified in the future. And the performance of EDA/DE-EIG can be exploited with more improvements. Exploration and exploitation [26] is an interest topic in evolutionary algorithms, and this paper is a fundamental study of the DE and EDA on this issue. More work can be researched on basis of this.

ACKNOWLEDGEMENT

This work is supported by China National Instrumentation Program under Grant No.2012YQ180132, the National Natural Science Foundation of China under Grant No.61273313, 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. Larranaga 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. 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] N. Hansen, A. Auger, S. Finck, and R. Ros, “real-parameter black-box optimization benchmarking 2012 :Experiment setup,” INRIA, France, Tech. Rep., 2012.

[21] 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.

[22] 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.

[23] 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.

[24] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, 1999.

[25] 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.

[26] M. Crepinsek, S.-H. Liu, and M. Mernik, “Exploration and exploitation in evolutionary algorithms: A survey,” *ACM COMPUTING SURVEYS*, vol. 45, no. 3, 2013.