Differential Evolution with Auto-enhanced Population Diversity: the Experiments on the CEC'2016 Competition

Ming Yang
School of Computer Science
China University of Geosciences
Wuhan, China 430074
Email: yangming0702@gmail.com

Jing Guan
China Ship Development and Design Center
Wuhan, China, 430064
Email: g_jing0414@163.com

Changhe Li
School of Computer Science
China University of Geosciences
Wuhan, China 430074
Email: changhe.lw@gmail.com

***Abstract*—For the differential evolution (DE) algorithms, there are many parameter adaptation methods, which aim at tuning the mutation factor F and the crossover probability CR. When the population diversity is very small and has been converged in a local optimum, even if the evolution goes on, the population will no longer improve. This is also true for the DE algorithms with adaptive F and CR. The enhancement of population diversity is necessary to DE algorithms. In this paper, we test the JADE algorithm with auto-enhanced population diversity (AEPD) on the newest benchmark functions.**

I. Introduction

Differential evolution (DE), introduced by Price and Storn [1], is a simple yet powerful evolutionary algorithm (EA) for global optimization problems. Nowadays DE has become one of the most frequently used EAs for solving global optimization problems [2], mainly because it has good convergence property and is easy to understand.

There are three control parameters in DE: the amplification factor of the difference vector—F, the crossover control parameter—CR, and the population size—NP. The control parameters involved in DE are highly dependent on the problems to be solved [3]–[5], and the original DE algorithm keeps all the three control parameters fixed during the optimization process [4]. For a specific task, it may have to spend a huge amount of time to try and fine-tune the corresponding parameters. To address this issue, several adaptive and self-adaptive DE algorithms regarding F and CR were developed to solve general problems efficiently [6]–[9].

Although adaptive DE algorithms have been proposed, they mainly focus on tuning the mutation factors F and crossover probabilities CR. DE is faced with the issues of stagnation and premature convergence [10]. If a population is stagnant, it is unable to generate a better child solution even though the population is not premature converged [11]. That is, even though the population diversity is not poor, the algorithm is still unable to find any better solution. If a population is premature converged, the individuals converge to local optima. Due to the loss of diversity, a premature converged population is unable to generate any better solution. Yang et al. found that the convergence characteristics of population are different in different dimensions [12]. To address the above issue of reducing the ineffective moves and to solve the problems of population premature convergence and stagnation, Yang et al. [12] proposed a novel idea, i.e., auto-enhanced population diversity (AEPD) to improve the performance of a general DE variant by diversifying a small size of population at the dimensional level.

In [12], Yang et al. tested AEPD on the CEC'2005 functions. In this paper, we show the experimental results of AEPD on the newest version of functions (i.e., CEC'2016 competition functions).

II. Auto-enhanced Population Diversity

In this section, we briefly introduce the auto-enhanced population diversity (AEPD) proposed in [12].

In [12], we have pointed out that "When a population gets trapped in a local optimum, individuals are almost the same. The difference vectors generated by the mutation strategies in DE are almost zero. Therefore, the mutation strategies cannot generate new vectors that are far beyond the location of the current population. As a result, the population cannot jump out of the local optimum. This is a fatal defect of DE." Therefore, an auto-enhancement of population diversity is necessary to DE.

Fig. 1 shows the changes of standard deviation (std) in each dimension for all the individuals. In this figure, the standard deviation value in dimension 1 (variable x_1) decreases to almost zero at the end of the run (i.e., $std_1 \approx 10^{-15}$). However, in the other dimensions, the standard deviation values of individual solutions at the end of the run are far from zero compared with the standard deviation of x_1 (i.e., $std_i \approx 10^{-4}$, i=2,...,5), which indicates that the population can be further improved in these dimensions. The experimental results on the *Sphere* function show that the evolutionary progress in each dimension is asynchronous with each other. Based on this finding, in [12], we detected whether the population is converging or stagnant at the dimensional level. The population is re-diversified only when it is converging or stagnant.

978-1-5090-0623-6/16/$31.00

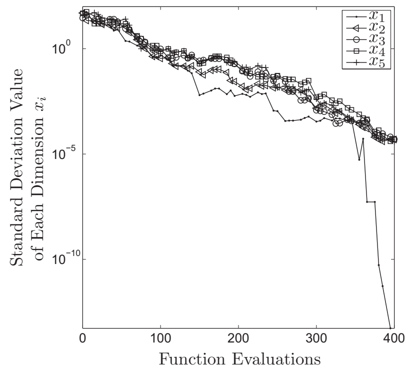

Fig. 1. Changes at the dimensional level of each individual in the DE/rand/1/bin algorithm on the 5-dimensional *Sphere* function in five dimensions with NP=5 in a typical run.

In order to measure population diversity in each dimension, the mean and the standard deviation of individuals' gene values in the j-th dimension at the G-th generation is calculated by:

$$m_{j,G} = \frac{1}{NP}\sum_{i=1}^{NP} x_{i,j,G}, \quad (1)$$

$$std_{j,G} = \sqrt{\frac{1}{NP}\sum_{i=1}^{NP}(x_{i,j,G} - m_{j,G})^2}, \quad (2)$$

where $m_{j,G}$ and $std_{j,G}$ are the mean and standard deviation of the population in the j-th dimension at generation G, respectively. NP is the population size. The value of $std_{j,G}$ is used to measure population diversity in the j-th dimension.

If the value of $std_{j,G}$ is smaller than a threshold, AEPD considers the population is converged in the j-th dimension. The threshold can be adaptive to the state of population convergence. The population can gradually converge to an optimum.

In the case of stagnation, the distribution of population does not change for ever. In practice if the distribution, the mean and the standard deviation of population, has not changed for several successive generations, AEPD considers the population is stagnant.

If the population starts converging or stagnating in the j-th dimension, AEPD diversifies the population in the j-th dimension. $x_{i,j,G+1}$, the j-th value of $\mathbf{x}_{i,G+1} = (x_{i,1,G+1}, x_{i,2,G+1}, \ldots, x_{i,D,G+1})$, $i \in [1, NP]$, is regenerated as follows:

$$x_{i,j,G+1} = low_{j,G} + (up_{j,G} - low_{j,G}) \cdot randn_{j,G} \quad (3)$$

where

$$low_{j,G} = \min(m_{j,G}, x_{low,j})^1 \quad (4)$$

$$up_{j,G} = \max(m_{j,G}, x_{up,j})^1; \quad (5)$$

[1]For unconstrained functions, $low_{j,G}$ and $up_{j,G}$ are allowed to be set beyond of the predefined bounds. This is helpful to solve functions whose global optima are outside of predefined range.

Algorithm 1 AEPD-JADE Algorithm

1: Set μ_{CR}=0.5, μ_F=0.5 and A=$\emptyset$;
2: Generate uniform random individuals for the initial population P_0;
3: Evaluate the fitness for each individual in P_0;
4: $G = 1$;
5: **while** the stop criterion is not satisfied **do**
6: S_F=$\emptyset$, S_{CR}=$\emptyset$;
7: **for** each individual $\mathbf{x}_{i,G} \in P_G$ **do**
8: Generate $CR_i = randn_i(\mu_{CR}, 0.1), F_i = randc_i(\mu_F, 0.1)$; /*$randn_i$ is a random number within the range [0,1] and follows a normal distribution. $randc_i$ is a random number within the range [0,1] and follows a Cauchy distribution. */
9: Randomly choose $\mathbf{x}^p_{best,G}$ as one of the $100p\%$ best vectors;
10: Randomly choose $\mathbf{x}_{r_1,G} \neq \mathbf{x}_{i,G}$ from the current population P_G;
11: Randomly choose $\tilde{\mathbf{x}}_{r_2,G} \neq \mathbf{x}_{r_1,G} \neq \mathbf{x}_{i,G}$ from $P_G \bigcup A$;
12: $\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F_i \cdot (\mathbf{x}^p_{best,G} - \mathbf{x}_{i,G}) + F_i \cdot (\mathbf{x}_{r_1,G} - \tilde{\mathbf{x}}_{r_2,G})$;
13: j_{rand}=rndint(1,D);
14: **for** j=1 to D **do**
15: **if** rnd(0,1)$<$ CR_i or $j == j_{rand}$ **then**
16: $u_{i,j,G} = v_{i,j,G}$;
17: **else**
18: $u_{i,j,G} = x_{i,j,G}$;
19: **end if**
20: **end for**
21: **if** $u_{i,j,G} \notin [x_{low,j}, x_{up,j}]$ **then**
22: Use Eq. (6) to map $u_{i,j,G}$ to be in the search range $[x_{low,j}, x_{up,j}]$;
23: **end if**
24: Evaluate the offspring $\mathbf{u}_{i,G}$;
25: **if** $\mathbf{u}_{i,G}$ is better than $\mathbf{x}_{i,G}$ **then**
26: $\mathbf{x}_{i,G+1} = \mathbf{u}_{i,G}$; $\mathbf{x}_{i,G} \rightarrow A$; $CR_i \rightarrow S_{CR}$; $F_i \rightarrow S_F$;
27: **else**
28: $\mathbf{x}_{i,G+1} = \mathbf{x}_{i,G}$;
29: **end if**
30: **end for**
31: Randomly remove solutions from A so that $|A| \leq NP$;
32: $\mu_{CR} = (1-c) \cdot \mu_{CR} + c \cdot mean_A(S_{CR})$;
33: $\mu_F = (1-c) \cdot \mu_F + c \cdot mean_L(S_F)$;
34: Implement the auto-enhancement of population diversity (AEPD) proposed in [12];
35: **if** *The population diversity is enhanced* **then**
36: *μ_{CR}=0.5, μ_F=0.5 and A=$\emptyset$;*
37: **end if**
38: $G = G + 1$;
39: **end while**

$x_{low,j}$ and $x_{up,j}$ are predefined lower and upper bounds for the j-th dimension, respectively; $randn_{j,G}$ is a random number within the range [0,1] and follows a normal distribution. The mean value of $randn_{j,G}$ is the mapping value of the best solution got by algorithms so far into the range [$x_{low,j}$, $x_{up,j}$].

The detailed introduction of AEPD can be seen in [12]. Algorithm 1 presents the pseudo-code of AEPD-JADE, which combines AEPD with the JADE [9] algorithm. If the population is regenerated, in this paper, we reset the values of μ_{CR}, μ_F and the archive A in JADE (see Steps 35 to 37 in Algorithm 1). If a trial vector $\mathbf{u}_i$ is generated outside of the search range in the j-th dimension after the mutation, in this paper, we map $u_{i,j}$ to the valid search range as follows:

$$u_{i,j} = \begin{cases} x_{low,j} + mod(u_{i,j} - x_{up,j}, x_{up,j} - x_{low,j}) & \text{if } u_{i,j} > x_{up,j} \\ x_{up,j} - mod(x_{low,j} - u_{i,j}, x_{up,j} - x_{low,j}) & \text{if } u_{i,j} < x_{low,j} \end{cases} \quad (6)$$

where $x_{low,j}$ and $x_{up,j}$ are the predefined lower and upper bounds, respectively.

TABLE I
THE COMPUTATIONAL COMPLEXITY OF AEPD-JADE. THE TIME UNIT IS SECOND.

	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
D=10	0.123	1.150	13.476	99.988
D=30	0.123	1.326	14.272	105.025
D=50	0.123	1.599	14.789	107.002

TABLE II
THE ERROR VALUES OF AEPD-JADE ON THE 10-DIMENSIONAL FUNCTIONS OVER 51 INDEPENDENT RUNS.

Func.	Best	Worst	Median	Mean	Std
1	0.00e+00	9.84e-04	0.00e+00	1.93e-05	1.36e-04
2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
4	0.00e+00	3.48e+01	3.48e+01	2.27e+01	1.64e+01
5	0.00e+00	2.01e+01	2.00e+01	1.65e+01	7.64e+00
6	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
7	0.00e+00	3.20e-02	4.49e-08	3.96e-03	6.48e-03
8	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
9	0.00e+00	3.98e+00	1.01e+00	1.34e+00	9.78e-01
10	0.00e+00	1.25e-01	0.00e+00	2.57e-02	3.54e-02
11	1.87e-01	2.40e+02	1.56e+01	4.02e+01	5.33e+01
12	7.04e-03	1.68e-01	7.42e-02	8.08e-02	3.94e-02
13	3.00e-02	1.08e-01	6.32e-02	6.37e-02	1.64e-02
14	1.53e-02	1.66e-01	5.14e-02	5.93e-02	3.12e-02
15	2.52e-01	6.27e-01	4.36e-01	4.33e-01	8.28e-02
16	9.54e-02	2.05e+00	1.00e+00	8.81e-01	5.05e-01
17	2.92e-04	1.34e+02	1.36e+01	3.06e+01	3.66e+01
18	9.82e-04	2.39e+00	4.48e-01	7.38e-01	7.54e-01
19	2.01e-03	1.19e-01	3.01e-02	3.52e-02	2.38e-02
20	9.15e-06	1.49e+00	7.84e-02	1.99e-01	2.78e-01
21	2.00e-06	1.68e+01	1.77e-02	4.68e-01	2.31e+00
22	5.48e-05	3.15e-01	7.24e-03	2.23e-02	5.97e-02
23	3.29e+02	3.29e+02	3.29e+02	3.29e+02	2.27e-13
24	1.00e+02	1.14e+02	1.07e+02	1.06e+02	4.08e+00
25	1.09e+02	2.00e+02	1.38e+02	1.48e+02	2.97e+01
26	1.00e+02	1.00e+02	1.00e+02	1.00e+02	1.50e-02
27	8.70e-01	4.00e+02	3.00e+02	2.35e+02	1.39e+02
28	3.57e+02	4.90e+02	3.57e+02	3.77e+02	3.75e+01
29	1.27e+02	2.35e+02	2.24e+02	2.15e+02	2.13e+01
30	3.35e+02	7.13e+02	5.42e+02	5.43e+02	6.93e+01

III. EXPERIMENTAL RESULTS

30 test instances, which were used in the CEC'2016 competition on single objective real-parameter numerical optimization, were used to study the performance of the AEPD-JADE algorithm. A detailed description of these test instances can be found in [13]. For each test function, 51 independent runs were conducted with $MaxFEs$=10000$\times D$ function evaluations as the termination criterion where D is the number of dimensions of the problems, which is suggested in [13]. We use the JADE algorithm with an archive in this paper since it showed promising results compared with JADE without an archive in [9]. The parameters c and p of JADE were set to 0.1 and 0.2, which were recommended in [9], respectively. The population size of AEPD-JADE was set to NP=20 for all experiments.

Table I summarizes the computational complexity of AEPD-JADE as suggested in [13]. It can be seen in Table I that the computational complexity of AEPD-JADE is not very sensitive to the dimensionality of functions. AEPD-JADE re-diversifies the population only when the population has been converged or stagnant.

TABLE III
THE ERROR VALUES OF AEPD-JADE ON THE 30-DIMENSIONAL FUNCTIONS OVER 51 INDEPENDENT RUNS.

Func.	Best	Worst	Median	Mean	Std
1	2.42e+00	5.42e+03	3.15e+02	7.04e+02	1.16e+03
2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
4	9.64e-05	2.93e-01	1.89e-02	3.04e-02	4.61e-02
5	2.00e+01	2.03e+01	2.01e+01	2.01e+01	5.22e-02
6	8.88e-03	5.78e+00	2.54e+00	2.56e+00	1.68e+00
7	0.00e+00	2.46e-02	0.00e+00	2.35e-03	5.58e-03
8	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
9	9.95e+00	3.88e+01	2.19e+01	2.20e+01	6.05e+00
10	0.00e+00	1.22e+00	8.33e-02	1.51e-01	2.66e-01
11	4.95e+02	2.15e+03	1.34e+03	1.37e+03	3.48e+02
12	1.58e-02	2.64e-01	1.20e-01	1.15e-01	6.82e-02
13	9.11e-02	2.25e-01	1.35e-01	1.41e-01	3.33e-02
14	7.21e-02	2.15e-01	1.45e-01	1.45e-01	3.44e-02
15	2.04e+00	5.31e+00	3.26e+00	3.35e+00	7.49e-01
16	7.11e+00	1.04e+01	8.73e+00	8.69e+00	7.27e-01
17	5.06e+01	7.68e+03	8.38e+02	1.22e+03	1.27e+03
18	3.56e+00	1.73e+02	4.29e+01	5.06e+01	3.73e+01
19	2.36e+00	6.29e+01	4.19e+00	5.24e+00	8.20e+00
20	4.62e+00	9.24e+01	3.36e+01	3.95e+01	2.36e+01
21	1.30e+02	1.10e+03	4.71e+02	4.91e+02	2.42e+02
22	2.10e+01	2.77e+02	1.45e+02	1.51e+02	5.30e+01
23	3.15e+02	3.15e+02	3.15e+02	3.15e+02	1.71e-13
24	2.22e+02	2.27e+02	2.24e+02	2.24e+02	9.69e-01
25	2.03e+02	2.14e+02	2.06e+02	2.07e+02	2.88e+00
26	1.00e+02	2.00e+02	1.00e+02	1.08e+02	2.68e+01
27	3.00e+02	5.42e+02	4.01e+02	3.87e+02	4.38e+01
28	7.42e+02	9.33e+02	8.44e+02	8.41e+02	3.90e+01
29	2.84e+02	1.10e+03	5.17e+02	5.57e+02	2.20e+02
30	5.80e+02	3.56e+03	1.57e+03	1.62e+03	6.37e+02

Tables II to V summarize the results over 51 independent runs on the 10-dimensional, 30-dimensional, 50-dimensional and 100-dimensional functions, respectively. Although AEPD-JADE has a fixed and small population size (i.e., NP=20), AEPD-JADE can get the global optimum for some unimodal functions (f_2 and f_3) and simple multimodal functions (f_7 and f_8) with different dimensionality.

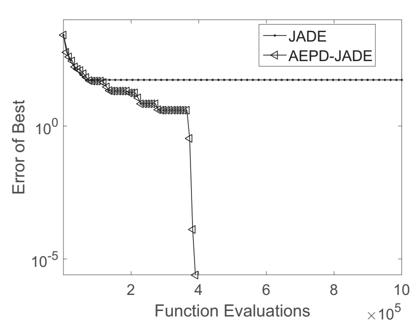

Fig. 2. The convergence graph of AEPD-JADE and JADE on 100-dimensional f_8 with NP=20 and the same initial population in a typical run.

TABLE IV
THE ERROR VALUES OF AEPD-JADE ON THE 50-DIMENSIONAL FUNCTIONS OVER 51 INDEPENDENT RUNS.

Func.	Best	Worst	Median	Mean	Std
1	7.94e+02	3.79e+04	8.18e+03	1.06e+04	7.85e+03
2	0.00e+00	3.57e-06	0.00e+00	1.89e-07	6.81e-07
3	0.00e+00	6.56e-04	0.00e+00	1.68e-05	9.20e-05
4	1.94e-04	9.81e+01	4.38e+00	1.02e+01	2.42e+01
5	2.01e+01	2.03e+01	2.01e+01	2.01e+01	6.49e-02
6	4.27e+00	1.59e+01	1.10e+01	1.06e+01	2.70e+00
7	0.00e+00	4.91e-02	7.40e-03	7.81e-03	1.07e-02
8	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
9	2.89e+01	9.25e+01	6.47e+01	6.46e+01	1.19e+01
10	8.74e-02	6.53e+00	8.45e-01	1.18e+00	1.15e+00
11	2.30e+03	5.02e+03	3.35e+03	3.42e+03	6.71e+02
12	2.29e-02	3.26e-01	7.52e-02	1.01e-01	7.84e-02
13	1.45e-01	3.64e-01	2.15e-01	2.21e-01	4.41e-02
14	1.44e-01	2.69e-01	1.95e-01	1.95e-01	2.76e-02
15	4.80e+00	1.26e+01	7.79e+00	7.78e+00	1.64e+00
16	1.60e+01	1.87e+01	1.73e+01	1.73e+01	8.04e-01
17	1.08e+03	1.28e+05	7.24e+03	1.07e+04	1.77e+04
18	1.13e+01	3.43e+03	2.33e+02	5.25e+02	7.49e+02
19	6.09e+00	7.61e+01	3.41e+01	3.82e+01	2.39e+01
20	2.78e+01	2.03e+02	1.20e+02	1.16e+02	4.18e+01
21	5.00e+02	2.98e+04	2.37e+03	4.07e+03	5.29e+03
22	2.16e+01	5.76e+02	2.78e+02	2.75e+02	1.39e+02
23	3.44e+02	3.44e+02	3.44e+02	3.44e+02	1.14e-13
24	2.55e+02	2.81e+02	2.71e+02	2.69e+02	6.06e+00
25	2.08e+02	2.28e+02	2.23e+02	2.22e+02	4.10e+00
26	1.00e+02	2.00e+02	1.00e+02	1.04e+02	1.94e+01
27	4.62e+02	8.01e+02	6.13e+02	6.17e+02	6.72e+01
28	1.19e+03	1.72e+03	1.34e+03	1.36e+03	1.06e+02
29	5.31e+02	1.48e+03	9.69e+02	9.79e+02	2.41e+02
30	8.54e+03	1.37e+04	1.03e+04	1.05e+04	1.24e+03

TABLE V
THE ERROR VALUES OF AEPD-JADE ON THE 100-DIMENSIONAL FUNCTIONS OVER 51 INDEPENDENT RUNS.

Func.	Best	Worst	Median	Mean	Std
1	3.03e+04	3.31e+05	1.27e+05	1.42e+05	6.13e+04
2	0.00e+00	1.36e+01	4.05e-06	7.19e-01	2.61e+00
3	0.00e+00	1.17e-02	7.95e-08	2.34e-04	1.62e-03
4	4.80e+00	1.86e+02	8.52e+01	9.29e+01	3.83e+01
5	2.01e+01	2.05e+01	2.02e+01	2.02e+01	9.47e-02
6	4.01e+01	6.96e+01	5.43e+01	5.45e+01	5.86e+00
7	0.00e+00	4.42e-02	0.00e+00	6.80e-03	1.07e-02
8	0.00e+00	8.95e+00	9.95e-01	1.42e+00	1.91e+00
9	2.02e+02	3.89e+02	2.76e+02	2.82e+02	4.19e+01
10	2.25e+00	1.24e+02	5.67e+00	7.86e+00	1.65e+01
11	8.10e+03	1.41e+04	1.05e+04	1.07e+04	1.21e+03
12	3.60e-02	5.18e-01	9.00e-02	1.39e-01	1.16e-01
13	2.77e-01	5.00e-01	3.62e-01	3.64e-01	4.67e-02
14	1.98e-01	6.48e-01	2.44e-01	2.53e-01	6.03e-02
15	2.06e+01	4.54e+01	3.03e+01	3.11e+01	6.22e+00
16	3.64e+01	4.16e+01	3.94e+01	3.92e+01	1.31e+00
17	3.06e+04	1.68e+05	7.88e+04	8.11e+04	2.83e+04
18	1.11e+02	5.89e+03	7.60e+02	1.35e+03	1.44e+03
19	1.79e+01	1.23e+02	6.01e+01	6.59e+01	2.57e+01
20	1.43e+02	5.18e+02	2.68e+02	2.72e+02	7.01e+01
21	7.04e+03	6.48e+04	1.92e+04	2.43e+04	1.39e+04
22	1.74e+02	1.51e+03	7.61e+02	8.45e+02	3.18e+02
23	3.48e+02	3.48e+02	3.48e+02	3.48e+02	1.71e-13
24	3.66e+02	3.91e+02	3.76e+02	3.78e+02	6.66e+00
25	2.54e+02	2.94e+02	2.74e+02	2.74e+02	9.60e+00
26	2.00e+02	2.00e+02	2.00e+02	2.00e+02	8.50e-03
27	1.23e+03	1.82e+03	1.51e+03	1.52e+03	1.37e+02
28	2.89e+03	4.78e+03	3.73e+03	3.73e+03	4.33e+02
29	1.06e+03	1.90e+03	1.47e+03	1.49e+03	1.92e+02
30	6.13e+03	1.16e+04	8.33e+03	8.40e+03	1.04e+03

Fig. 2 shows the convergence behavior of JADE and AEPD-JADE on 100-dimensional f_8, on which AEPD can represent a typical behavior, when the two algorithms have a same initial population (i.e., NP=20). Because the population size is small, JADE converged fast in the early stage of evolution. After about 1×10^5 function evaluations, JADE converged to a local optimum and could not jump out of this local optimum. For JADE, although the parameters (the difference vector F and the crossover probability CR) are adaptive, JADE cannot get a better solution because the population diversity is so poor. But AEPD-JADE can get a better solution as the evolution progresses. At about 4×10^5 function evaluations, the error value of the best solution got by AEPD-JADE was 0. AEPD-JADE got the global best solution. This is because AEPD can re-diversify the population when the population diversity is poor.

IV. Conclusion

In this paper, we show the experimental results of AEPD-JADE on the newest test functions (i.e., the CEC'2016 competition functions). Although the population size of AEPD-JADE, which was proposed by us before, is small (i.e. NP=20) and fixed, AEDP-JADE can get good results on some functions with different dimensionality. AEPD helped the algorithm become less sensitive to population size, a parameter widely considered problem dependent for many DE algorithms.

Acknowledgment

Thanks for the funding of the National Natural Science Foundation of China (Grand Nos. 61305086, 61203306 and 61305079).

References

[1] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Opt.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
[2] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag, 2005.
[3] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," in *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, 2002, pp. 293–298.
[4] J. Liu and J. Lampinen, "On setting the control parameter of the differential evolution method," in *Proc. 8th Int. Conf. Soft Computing*, 2002, pp. 11–18.
[5] R. Mallipeddi and P. Suganthan, "Empirical study on the effect of population size on differential evolution algorithm," in *Proc. IEEE CEC*, Jue. 2008, pp. 3663–3670.
[6] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput. A Fusion Found., Methodol. Applicat.*, vol. 9, no. 6, pp. 448–462, 2005.
[7] A. Qin and P. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE CEC*, 2005, pp. 1785–1791.

[8] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, 2006.
[9] J. Zhang and C. Arthur, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, 2009.
[10] N. Padhye, P. Mittal, and K. Deb, "Differential evolution: Performances and analyses," in *Proc. IEEE Congr. Evol. Comput.*, 2013, pp. 1960–1967.
[11] J. Lampinen and I. Zelinka, "On stagnation of the differential evolution algorithm," in *Proc. MENDEL, 6th International Mendel Conference on Soft Computing*, 2000, pp. 76–83.
[12] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *Cybernetics, IEEE Transactions on*, vol. 45, no. 2, pp. 302–315, Feb 2015.
[13] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore, Tech. Rep. 201311, 2013.