NEW VARIANTS OF ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHM WITH COMPETING STRATEGIES

Petr BUJOK*, Josef Tvrdík**
*Department of Informatics and Computers, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic, tel. +420 59709 2147, E-mail: petr.bujok@osu.cz
**Department of Computer Sciences and Centre of Excellence IT4Innovations, Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, 30. dubna 22, 701 03 Ostrava, Czech Republic, tel. +420 59709 2183; E-mail: josef.tvrdik@osu.cz

ABSTRACT

New variants of the adaptive competitive differential evolution algorithm are proposed and tested experimentally on the CEC 2013 test suite. In the new variants, the adaptation is based on the competition of several strategies. The current-to-pbest mutation borrowed from JADE is included into the pool of the competing strategies in newly proposed variants. The aim of the experimental comparison is to find whether the presence of the current-to-pbest mutation strategy increases the efficiency of the differential evolution algorithm, especially on rotated objective functions. The results of the experiments show that the new variants performed better in a few of the test problems, while the benefit is not observed in the majority of the test problems.

Keywords: *differential evolution, current-to-pbest mutation, strategy competition, CEC 2013 test suite.*

1. INTRODUCTION

This paper is an extended version of the conference submission [2]. Compared to [2], two other pools of competing strategies are proposed and all the variants of the algorithm are compared experimentally on the CEC 2013 benchmark suite [5].

Differential evolution (DE) proposed in [9] is a population-based optimization algorithm for single-objective problems with a real-valued objective function. The possible solutions are represented as vectors with real-number components, $\vec{x} = (x_1,\ x_2,\ ,\ldots,x_D)$, D is the dimension of the problem. The population is placed in the search space $\Omega = \prod_{j=1}^{D}[a_j,\ b_j]$, $a_j < b_j$, $j = 1,\ 2,\ ,\ldots,D$ and evolves during the search to the state of higher fitness. The solution of the problem is the global minimum point $\vec{x}^*$ satisfying condition $f(\vec{x}^*) \le f(\vec{x})$, $\forall \vec{x} \in \Omega$.

Algorithm 1 Differential evolution algorithm

```
initialize population P = {x1, x2, ..., xN}
while stopping condition not reached do
   for i = 1,2,...,N do
      create a new trial vector y
      if f(y) ≤ f(xi) then
         insert y into Q
      else
         insert xi into Q
      end if
   end for
   P ← Q
end while
```

The population of size N is developed step-by-step from a generation P to a generation Q by application of evolutionary operators, i.e. mutation, crossover, and selection. The basic scheme of DE algorithm written in pseudo-code is shown in Algorithm 1. The new trial point is created from a mutant point $\vec{u}$ generated by using a kind of mutation and from the current point of the population by the application of the crossover. Fitter point from the pair of $(\vec{x}_i,\ \vec{y})$, based on the value of the objective function, is selected to the new generation Q.

The DE algorithm has been studied intensively in recent period. Comprehensive summary of advanced results in DE research is available in [7] and [3]. Several kinds of mutation and crossover were suggested as well as some adaptive or self-adaptive DE variants. The main goal of designing adaptive variants of DE is to enable the adaptation of the search carried out during the run of the DE algorithm to the current problem to be solved.

The new test suite of 28 functions was proposed for the special session on Real-Parameter Optimization, a part of Congress on Evolutionary Computation (CEC) 2013. This session was held as a competition of stochastic single-objective optimization algorithms. The functions are described in the report [5], including the experimental settings required for the competition. The source code of the functions is also available at the web site given in the report. The benchmark functions can be used at several levels of problem dimension varying from 2 to 100. We can expect that this test suite will become one of the most relevant benchmark required for publishing new single-objective optimization algorithms.

We took part in the CEC 2013 special session mentioned above with the paper [13], where an adaptive version of differential evolution based on the competition of DE strategies was applied [12]. Our DE variant was ranked in the first half of 21 compared algorithms with respect to their efficiency. This DE variant performs well on the problems, where the objective function is not rotated, whilst the performance in the problems with rotated functions is worse. Similar difficulties with the rotated functions occurred in all DE variants taking part in the CEC 2013 competition including the best performing DE variant of SHADE [10].

In this paper, novel variants of the competitive DE combining two adaptive approaches are proposed and compared experimentally with the "parental" algorithms [12, 18] on the CEC 2013 test suite.

The rest of the paper is organized as follows. In Section 2 the "parental" algorithms are described. Three new adaptive DE variants are proposed in Section 3. Experimental setup is defined in Section 4. The results are presented in Section 5 and the last section concludes the paper.

2. SOME ADAPTIVE VARIANTS OF DIFFERENTIAL EVOLUTION

It is know that standard DE can be a very efficient optimization algorithm but the efficiency is strongly dependent on the setting of the control parameters F and CR for the problem to be solved. The tuning of the control parameters by trial-and-error method is time-consuming. Hence many adaptive or self-adaptive DE variants have been proposed in last decade.

Seven adaptive DE variants [1, 6, 8, 12, 15, 18] were experimentally compared on six standard benchmark functions at three levels of dimension in [14]. It was found that JADE [18] and *b6e6rl* [12] were the best performing algorithms in the comparison, JADE was the fastest and the second reliable in average, while the *b6e6rl* was the most reliable and the second in convergence speed. That is why these "parental" algorithms are exploited in the proposed new variants.

2.1. JADE

JADE variant of adaptive differential evolution [18] extends the original DE concept with three different improvements – current-to-pbest mutation, a new adaptive control of parameters F and CR, and archive. The mutant vector $\vec{u}$ is generated in the following manner:

$$\vec{u} = \vec{x}_i + F\,(\vec{x}_{\text{pbest}} - \vec{x}_i) + F\,(\vec{x}_{r1} - \vec{x}_{r2}), \qquad (1)$$

where $\vec{x}_{\text{pbest}}$ is randomly chosen from $100\,p\,\%$ best individuals with input parameter $p = 0.05$ recommended in [18]. The vector $\vec{x}_{r1}$ is randomly selected from P ($r1 \neq i$), $\vec{x}_{r2}$ is randomly selected from the union $P \bigcup A$ ($r2 \neq i \neq r1$) of the current population P and the archive A. In every generation, parent individuals replaced by better offspring individuals are put into the archive and the archive size is reduced to N individuals by randomly dropping surplus individuals. The trial vector is generated from $\vec{u}$ and $\vec{x}_i$ using the binomial crossover. CR and F are independently generated for each individual $\vec{x}_i$, CR is generated from the normal distribution of mean μ_{CR} and standard deviation 0.1, truncated to $[0, 1]$. F is generated from Cauchy distribution with location parameter μ_F and scale parameter 0.1, truncated to 1 if $F > 1$ or regenerated if $F < 0$, see [18] for details of μ_{CR} and μ_{F} adaptation.

2.2. Competitive DE

Competitive DE uses H strategies with their control-parameter values held in the pool [11, 12]. Any of H strategies can be chosen to create a new trial point $\vec{y}$. A strategy is selected randomly with probability q_h, $h = 1, 2, \ldots, H$. The values of probability are initialized uniformly, $q_h = 1/H$, and they are modified according to the success rate in the preceding steps. The hth strategy is considered successful if it produces a trial vector entering into next generation. Probability q_h is evaluated as the relative frequency of success according to

$$q_h = \frac{n_h + n_0}{\sum_{j=1}^{H}(n_j + n_0)}, \qquad (2)$$

where n_h is the current count of the hth setting successes, and $n_0 > 0$ is an input parameter. The setting of $n_0 > 1$ prevents from a dramatic change in q_h by one random successful use of the hth strategy. To avoid degeneration of the search process, the current values of q_h are reset to their starting values if any probability q_h decreases below some given limit δ, $\delta > 0$.

We use a variant of competitive DE that appeared well-performing and robust in different benchmark tests [12]. In this variant, denoted *b6e6rl* hereafter, 12 strategies are in competition ($H = 12$), six of them using the binomial crossover, rest of them using the exponential crossover.

The randrl/1 mutation (3) is applied in all the strategies, two different values of control parameter F are used, $F = 0.5$ and $F = 0.8$.

$$\vec{u} = \vec{r}_1^{\,x} + F\,(\vec{r}_2^{\,x} - \vec{r}_3^{\,x}), \qquad (3)$$

where the point $\vec{r}_1^{\,x}$ is tournament best among $\vec{r}_1$, $\vec{r}_2$, and $\vec{r}_3$, i.e. $f(\vec{r}_1^{\,x}) \leq f(\vec{r}_j^{\,x}),\ j = 2, 3$, as proposed in [4].

The mutation according to (3) can cause that a mutant point $\vec{u}$ moves out of the domain Ω. In such a case, the values of $u_j \notin [a_j, b_j]$ are turned over into Ω by using transformation $u_j \leftarrow 2a_j - u_j$ or $v_j \leftarrow 2b_j - u_j$ for the violated component. The same treatment of the mutation points escaping the Ω is also used in newly proposed algorithms.

The binomial crossover uses three different values of CR, $CR \in \{0, 0.5, 1\}$. The values of CR for the exponential crossover are evaluated from given values of mutation probability p_m as real roots of polynomial equation [17]

$$CR^D - D\,p_m\,CR + D\,p_m - 1 = 0. \qquad (4)$$

Three values of p_m used in this DE variant are set up equidistantly in the interval $(1/D, 1)$. Details of the CR setting for the exponential crossover can be found e.g. in [14].

3. NEWLY PROPOSED VARIANTS OF COMPETITIVE DE

The adaptive mechanism based on the competition of strategies described in Section 2.2 is applied in all the newly proposed adaptive DE variants. The new variants differs only in the combination of DE strategies available in the pools from which the strategies are selected. Some strategies in the pools of competing strategies are derived from JADE, especially they exploit the current-to-pbest mutation.

3.1. b6e6pbest

This adaptive variant of DE (denoted *b6e6pbest* hereafter) is similar to *b6e6rl* but the mutation randrl/1 is replaced by the current-to-pbest mutation used in JADE. It is expected that application of the current-to-pbest mutation can help in the solution of rotated functions. An archive from JADE storing the old best solutions is also applied. The new algorithm is shown in pseudo-code in Algorithm 2.

Algorithm 2 Competitive DE algorithm b6e6pbest

initialize population $P = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$
initialize empty archive A of the size N
initialize probabilities of strategies
while stopping condition not reached **do**
  for $i = 1, 2, \ldots, N$ **do**
    choose a strategy by a roulette selection
    create a new trial vector $\vec{y}$
    if $f(\vec{y}) \le f(\vec{x}_i)$ **then**
      insert $\vec{y}$ into Q
      insert $\vec{x}$ into A
      update the value of probability of used strategy
    else
      insert $\vec{x}_i$ into Q
    end if
  end for
  $P \leftarrow Q$
end while

The population P of size N is initialized randomly uniformly distributed in the area of the possible solutions. In addition, the empty archive A of the size N for the storage the old solutions is also initialized. When the new trial point is inserted into next generation Q, the old solution $\vec{x}_i$ is stored in the archive A. If the archive is full, a randomly selected point in the A is replaced by the current $\vec{x}_i$. The parameters controlling the competition of strategies are set to the recommended values: $\delta = 1/(5 \times H)$, $n_0 = 2$ and the control parameter of mutation $p = 0.05$.

3.2. b3e3j6-F05

Twelve DE strategies are included into the competition in this algorithm (denoted *b3e3j6-F05* hereafter), six strategies use the randrl mutation like the b6ee6rl algorithm and in the other strategies the current-to-pbest mutation is used. Mutation parameter F is set up to 0.5 in all the twelve DE strategies. This setting is supposed to be helpful by more intensive search in the neighborhood of the current point. The *CR* parameters for the both types of the crossover are set up to the same values applied in the b6e6rl.

3.3. b3e3j6-F05F08

Like in the algorithms described before, twelve DE strategies are also included into the competition of the strategies in this algorithm labeled *b3e3j6-F05F08* hereafter. Six strategies use the randrl mutation with $F = 0.5$, three of them in combination with the binomial crossover, three of them with the exponential crossover. The other strategies use the current-to-pbest mutation with $F = 0.8$ combined with the binomial and exponential crossover. The higher value of F in half of the competing strategies should keep the population more dispersed compare to *b3e3j6-F05*. The *CR* parameters for the both types of the crossover are also set up to the values applied in the b6e6rl.

4. EXPERIMENTS

The aim of the experiments is to compare the performance of the proposed variants with the parental JADE and b6e6rl algorithms. The algorithms are implemented in Matlab 2010a and this environment was used for experiments. Experimental setting follows the requirements given in the report [5], where the suite of 28 benchmark minimization problems is also defined. The function values $f(\vec{x}^*)$ are also given in [5]. Thus, the obtained value of the function error $f_{\min} - f(\vec{x}^*)$ can be calculated for each run, where $f_{\min}$ is the minimum function value in the population at the end of the search. The source code of the test functions in C was downloaded from the web page given in [5] and compiled by Lcc-win32 C 2.4.1 compiler. Search range (domain) for all the test functions is $[-100, 100]^D$.

The tests were carried out at two levels of dimension, $D = 10$ and $D = 30$. For each test problem, 51 repeated runs were performed. The run stops if the prescribed value of MaxFES $= D \cdot 10^4$ is reached or if the minimum function error in the population is less than 10^{-8} because such a value of the error is considered sufficient for an acceptable approximation of the correct solution. The values of the function error less than 10^{-8} are treated as zero in further processing.

The population size was set up to $N = 100$ for all the algorithms and the problem dimension. The remaining control parameters of the algorithms were set up to the recommended values described in Section 2 and 3.

5. RESULTS

The basic characteristics of the experimental comparison of the algorithms are presented in Tables 3-11. The structure of the characteristics follows the requirements given in Report [5]. The values of characteristics for each problem are counted from 51 repeated runs. The values of the function errors less than 10^{-8} are substituted by zero in all the tables.

The efficiency of the five algorithms expressed by the values of the function error found in each of 51 runs was compared statistically by Kruskal-Wallis non-parametric analysis of variance. Kruskal-Wallis multiple comparison of the algorithms was applied to the results of the problems where a significant difference among the algorithms was found. The results of the comparison are shown in Table 1. If there is an algorithm significantly better than the others, it is evaluated as the winner in the corresponding problem. If there are two or more algorithms on the winning position and these algorithms are not different significantly, the wining position is shared by all of them, ordered in the decreasing sequence of their performance.

The counts of wins and shared wins across all the problems are summarized in Table 2, the problems with the no significant difference are not taken into account. Based on the results in Table 2, we can conclude that JADE is the best performing algorithm most frequently but each algorithm in the comparison is winning in some problems at least once and the shared wins were obtained several times by each algorithm. Among the newly proposed DE variants, b6e6pbest performs best in average (20 wins or shared wins out of 56 problems).

Table 1 Comparison of algorithm performance by Kruskal-Wallis test – best performing algorithms.

F	D = 10	D = 30
1	b6e6pbest, b6e6rl, JADE	b6e6rl, JADE, b6e6pbest
2	b6e6pbest, b6e6rl, JADE	JADE
3	b6e6pbest	b3e3j6F05F08
4	b6e6pbest, b6e6rl, JADE	b6e6pbest
5	b6e6pbest, b6e6rl, JADE	b6e6rl, JADE, b6e6pbest
6	b6e6rl	JADE, b6e6pbest
7	b3e3j6F05	JADE
8	No significant difference	All except b3e3j6F05
9	b3e3j6F05, JADE, b6e6pbest	JADE
10	JADE	b3e3j6F05F08, b6e6rl
11	b6e6pbest, b6e6rl, JADE	JADE, b6e6rl, b6e6pbest
12	JADE	JADE
13	JADE, b3e3j6F05	JADE
14	JADE, b6e6rl	b6e6rl, JADE
15	JADE	JADE
16	b3e3j6F05	b6e6rl, JADE, b3e3j6F05
17	No significant difference	All except b3e3j6F05
18	JADE	JADE
19	JADE	JADE
20	b3e3j6F05, JADE, b6e6pbest	JADE
21	b6e6rl, b3e3j6F05F08	All except JADE
22	JADE	JADE
23	JADE	JADE
24	b3e3j6F05	b3e3j6F05F08, JADE, b3e3j6F05
25	b3e3j6F05	b3e3j6F05, b6e6pbest, b3e3j6F05F08
26	JADE	JADE, b6e6pbest
27	b3e3j6F05, b6e6pbest	b6e6pbest, b3e3j6F05
28	b3e3j6F05F08, b6e6rl	No significant difference

Table 2 Counts of the best and the shared best positions according to Kruskal-Wallis multiple comparison.

D	Algorithm	#Best	#Shared
$D = 10$	JADE	8	9
	b6e6rl	1	8
	b6e6pbest	1	8
	b3e3j6-F05	4	4
	b3e3j6-F05F08	0	2
$D = 30$	JADE	11	10
	b6e6rl	0	9
	b6e6pbest	1	10
	b3e3j6-F05	0	5
	b3e3j6-F05F08	1	6

None of the five algorithms copes well with all the test problems. There are problems, where the error of the best solution found by the algorithm in the prescribed number of the function evaluations has the magnitude of 10^2, see Tables 3-11. Especially for the composition functions (problems F21 to F28) no algorithm tested here is able to find a better solution. It is not surprising because the composite functions are very difficult tasks for all optimization algorithms. Moreover, the performance of DE algorithms in some problems with rotated objective function (F2 - F4, F6 - F10, F12, F13, F15, F16, F18, F20, F21, F23 - F28) is not satisfactory.

6. CONCLUSIONS

The experimental comparison showed that newly proposed variants of the competitive DE algorithm do not outperformed the "parental" JADE algorithm, when average performance on the CEC 2013 suite problems is taken into account. Among the newly proposed DE variants, b6e6pbest performs best in average (20 wins or shared wins out of 56 test problems), while JADE achieved 38 wins including shared wins.

However, there are optimization problems, where some newly proposed DE variants performed well. Each of the newly proposed algorithm wins at least in one problem of the CEC 2013 test suite and the shared wins are obtained several times by each algorithm. It indicates that for a specified optimization problem a special combination of competing DE strategies in the pool is more convenient for the convergence than other combination. Such behavior of the algorithms found experimentally is in agreement with the results of No-Free-Lunch theorem [16] but its benefit of the result is limited for application in the solution of real-world optimization problems.

It was found that all the tested DE variants do not perform well on the most of the problems with rotated objective function. The inclusion of current-to-pbest strategy into the competitive adaptive DE does not bring sufficient enhancement of the performance in these problems. Thus, the proposal of an innovated algorithm with the pool of strategies increasing the efficiency of DE on rotated functions remains the challenge for further research.

ACKNOWLEDGEMENT

This work was supported by University of Ostrava from the project SGS15/PřF/2014 and by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

REFERENCES

[1] BREST, J., GREINER, S., BOŠKOVIČ, B., MERNIK, M., AND ŽUMER, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. In *IEEE Transactions on Evolutionary Computation* (2006), vol. 10, pp. 646–657.

[2] BUJOK, P., AND TVRDÍK, J. A new variant of competitive differential evolution algorithm. In *INFORMATICS'2013 - International Scientific Conference on Informatics* (Košice, 2013), FEEI TU of Košice, pp. 304–308.

[3] DAS, S., AND SUGANTHAN, P. N. Differential evolution: A survey of the state-of-the-art. In *IEEE Transactions on Evolutionary Computation* (2011), vol. 15, pp. 27–54.

[4] KAELO, P., AND ALI, M. M. A numerical study of some modified differential evolution algorithms. In *European J. Operational Research* (2006), vol. 169, pp. 1176–1184.

[5] LIANG, J. J., QU, B., SUGANTHAN, P. N., AND HERNANDEZ-DIAZ, A. G. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, 2013.

[6] MALLIPEDDI, R., SUGANTHAN, P. N., PAN, Q. K., AND TASGETIREN, M. F. Differential evolution algorithm with ensemble of parameters and mutation strategies, 2011.

[7] NERI, F., AND TIRRONEN, V. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review 33* (2010), 61–106.

[8] QIN, A. K., HUANG, V. L., AND SUGANTHAN, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. In *IEEE Transactions on Evolutionary Computation* (APR 2009), vol. 13, pp. 398–417.

[9] STORN, R., AND PRICE, K. V. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. In *J. Global Optimization* (1997), vol. 11, pp. 341–359.

[10] TANABE, R., AND FUKUNAGA, A. Success-history based parameter adaptation for differential evolution. In *2013 IEEE Conference on Evolutionary Computation* (Cancun, Mexico, June 20–23 2013), L. G. de la Fraga, Ed., vol. 1, pp. 71–78.

[11] TVRDÍK, J. Competitive differential evolution. In *MENDEL 2006, 12th International Conference on Soft Computing* (Brno, 2006), R. Matoušek and P. Ošmera, Eds., University of Technology, pp. 7–12.

[12] TVRDÍK, J. Self-adaptive variants of differential evolution with exponential crossover. In *Analele of West University Timisoara, Series Mathematics-Informatics* (2009), vol. 47, pp. 151–168.

[13] TVRDÍK, J., AND POLAKOVA, R. Competitive Differential Evolution Applied to CEC 2013 Problems. In *IEEE Congress on Evolutionary Computation 2013* (2013), IEEE Computational Intelligence Society, pp. 1651–1657.

[14] TVRDÍK, J., POLÁKOVÁ, R., VESELSKÝ, J., AND BUJOK, P. Adaptive variants of differential evolution: Towards control-parameter-free optimizers. In *Handbook of Optimization – From Classical to Modern Approach*, I. Zelinka, V. Snášel, and A. Abraham, Eds., Intellingent Systems Reference Library, vol. 38. Springer-Verlag, Berlin Heidelberg, 2012, pp. 423–449.

[15] WANG, Y., CAI, Z., AND ZHANG, Q. Differential evolution with composite trial vector generation strategies and control parameters. In *IEEE Transactions on Evolutionary Computation* (2011), vol. 15, pp. 55–66.

[16] WOLPERT, D. H., AND MACREADY, W. No free lunch theorems for optimization. In *IEEE Transactions on Evolutionary Computation* (1997), vol. 1, pp. 67–82.

[17] ZAHARIE, D. Influence of crossover on the behavior of differential evolution algorithms. In *Applied Soft Computing* (2009), vol. 9, pp. 1126–1138.

[18] ZHANG, J., AND SANDERSON, A. C. JADE: Adaptive differential evolution with optional external archive. In *IEEE Transactions on Evolutionary Computation* (2009), vol. 13, pp. 945–958.

Received May 11, 2014, accepted July 7, 2015

BIOGRAPHY

Petr Bujok was born on 30. 8. 1983. In 2008 he graduated (Mgr) with distinction at the Department of Informatics and Computers of the Faculty of Science at University of Ostrava. He defended his PhD in the field of parallel models of optimization techniques in 2013; his thesis title was "Parallelism in adaptive evolutionary algorithms". Since 2013 he is working as a assistant professor at the Department of Informatics and Computers. His scientific research is focusing on adaptive variants of evolutionary algorithms and their parallel models. In addition, he also deal with the statistical data processing.

Josef Tvrdík was born on 28. 6. 1943. He has been working at the Department of Informatics and Computers of University of Ostrava since 1990, since 2004 as Assoc. Prof. His scientific research is focusing on adaptive evolutionary algorithms and their application in computational statistics. He also deals with the biomedical applications of statistics.

Table 3 Basic Characteristics of function error, b6e6rl, $D = 10$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	0
2	0	0	0	0	0
3	0	6.31503	1.86E-02	2.87E-01	1.2313
4	0	0	0	0	0
5	0	0	0	0	0
6	0	9.81242	0	1.3468	3.41021
7	1.38E-02	1.81E-01	5.95E-02	6.54E-02	3.77E-02
8	20.1802	20.503	20.3583	20.3593	7.49E-02
9	9.86E-01	6.19913	4.85471	4.62035	9.37E-01
10	1.30E-02	1.42E-01	9.82E-02	9.76E-02	2.95E-02
11	0	0	0	0	0
12	5.97046	16.3914	12.3216	11.9187	2.63228
13	5.26893	20.4302	13.3977	13.4186	3.9391
14	0	6.25E-02	0	9.80E-03	2.29E-02
15	392.73	1113.48	882.615	849.553	158.593
16	5.89E-01	1.56243	1.06043	1.04941	1.97E-01
17	10.1224	10.1224	10.1224	10.1224	1.26E-14
18	21.8359	37.55	30.6852	30.7071	3.15786
19	2.93E-01	5.74E-01	4.26E-01	4.34E-01	6.25E-02
20	1.91115	3.42677	2.69786	2.64982	2.98E-01
21	200	400.194	400.194	368.791	73.5293
22	5.01559	102.082	18.3968	20.4875	17.4553
23	381.812	1193.88	918.923	886.649	162.68
24	111.881	209.192	206.295	200.658	18.0154
25	114.992	205.743	200.109	197.841	16.6632
26	106.565	200.017	200.017	156.718	44.6596
27	300.005	473.986	300.039	303.462	24.3558
28	100	300	300	268.628	73.458

Table 4 Basic Characteristics of function error, JADE, $D = 10$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	0
2	0	0	0	0	0
3	0	1519.65	8.69072	65.3855	218.05
4	0	8098.3	0	218.682	1203.94
5	0	0	0	0	0
6	0	9.81242	9.81242	6.92641	4.51547
7	6.14E-12	9.45E-01	2.25E-02	1.00E-01	1.77E-01
8	20.1364	20.4517	20.3788	20.3607	7.47E-02
9	1.45291	5.29509	3.8035	3.83377	8.22E-01
10	5.10E-03	4.32E-02	1.91E-02	2.00E-02	9.22E-03
11	0	0	0	0	0
12	2.27093	7.24048	4.43965	4.43427	1.23165
13	1.07571	10.7264	4.55144	4.95621	2.29185
14	0	6.25E-02	0	4.90E-03	1.70E-02
15	243.289	740.153	477.817	492.932	115.477
16	7.22E-01	1.5008	1.08145	1.11877	2.13E-01
17	10.1224	10.1224	10.1224	10.1224	1.26E-14
18	15.0662	22.4264	18.4838	18.34651	1.71822
19	2.57E-01	4.01E-01	3.34E-01	3.38E-01	3.67E-02
20	1.49735	3.18457	2.27887	2.29143	4.40E-01
21	200	400.194	400.194	396.267	28.0328
22	9.66E-08	100.001	1.26786	5.93799	14.266
23	220.144	842.444	448.01	480.143	145.998
24	108.49	210.404	200.151	198.064	18.2606
25	128.095	206.964	200.807	199.924	10.8532
26	101.575	200.017	107.198	136.021	43.7897
27	300	301.258	300.075	300.167	2.48E-01
28	100	300	300	296.078	28.0056

Table 5 Basic Characteristics of function error, b6e6pbest, $D = 10$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	0
2	0	0	0	0	0
3	0	7.14E-02	0	3.69E-03	1.42E-02
4	0	0	0	0	0
5	0	0	0	0	0
6	0	9.81242	9.81242	5.77201	4.87727
7	1.71E-04	8.71E-02	9.16E-03	1.31E-02	1.56E-02
8	20.1905	20.5061	20.3834	20.3698	6.77E-02
9	8.81E-02	6.26927	4.24983	3.93156	1.32163
10	8.14E-03	1.07E-01	4.65E-02	4.78E-02	1.78E-02
11	0	0	0	0	0
12	9.30E-01	9.40186	6.41642	6.48988	1.61251
13	3.38757	11.4131	6.83866	6.9651	1.88697
14	5.71E-05	2.17E-01	6.36E-02	7.44E-02	5.53E-02
15	430.339	992.613	754.327	753.255	143.513
16	5.27E-01	1.32219	9.24E-01	9.40E-01	1.92E-01
17	7.77E-03	10.1224	10.1224	9.92407	1.41633
18	16.9116	28.6776	24.2728	23.7399	2.49952
19	2.99E-01	5.29E-01	4.00E-01	4.03E-01	4.91E-02
20	1.6708	3.12941	2.28349	2.31911	3.55E-01
21	200	400.194	400.194	392.343	39.2459
22	7.87E-01	64.6129	15.2422	19.9918	13.8066
23	130.891	1078.12	769.183	756.324	182.338
24	200	208.648	200.002	201.629	2.98218
25	108.444	210.757	200.007	197.137	17.4591
26	102.769	200.017	200.017	156.002	47.1613
27	300	488.129	300	310.663	43.0999
28	100	300	300	292.157	39.2078

Table 6 Basic Characteristics of function error, b3e3j6-F05F08, $D = 10$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	1.40E-09
2	0	0	0	0	1.24E-09
3	0	6.31503	1.71E-06	3.74E-01	1.50003
4	0	0	0	0	1.72E-09
5	0	1.00E-08	0	0	1.18E-09
6	0	9.81242	0	1.1544	3.19292
7	3.64E-03	2.20E-01	2.57E-02	3.76E-02	3.83E-02
8	20.1802	20.4784	20.3738	20.3582	7.81E-02
9	5.08E-05	5.9947	4.54293	4.1065	1.57131
10	3.71E-02	1.69E-01	8.21E-02	8.47E-02	2.9E-02
11	0	0	0	0	1.27E-09
12	5.4376	15.6135	10.8879	10.5908	2.53997
13	4.05796	20.0459	11.2484	11.7068	3.53416
14	0	1.25E-01	9.30E-09	1.71E-02	3.08E-02
15	411.039	1191.54	817.837	825.65	164.593
16	5.75E-01	1.42941	1.04439	1.02629	1.79E-01
17	10.1224	10.1224	10.1224	10.1224	1.26E-14
18	24.0478	38.78	30.8158	31.0845	3.48422
19	1.73E-01	5.44E-01	4.31E-01	4.29E-01	6.53E-02
20	2.07821	3.0021	2.40051	2.47522	2.34E-01
21	200	400.194	400.194	376.642	65.1423
22	8.84177	105.327	15.7225	19.7737	17.0032
23	296.241	1171.94	813.22	793.93	166.582
24	109.178	210.012	200.024	197.523	21.9362
25	200.005	207.393	200.029	200.977	1.95775
26	105.397	200.017	116.914	153.253	44.6511
27	300.003	433.417	300.013	306.553	26.687
28	100	300	300	264.706	77.0027

Table 7 Basic Characteristics of function error, b3e3j6-F05, $D = 10$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	1.34E-09
2	0	0	0	0	1.63E-09
3	0	7.14E-02	9.27E-05	5.71E-03	1.69E-02
4	0	0	0	0	1.69E-09
5	0	0	0	0	1.53E-09
6	0	9.81242	0	4.23281	4.90813
7	2.01E-06	3.56E-02	8.79E-04	2.71E-03	5.33E-03
8	20.1423	20.5075	20.3589	20.3542	6.58E-02
9	4.52E-05	5.62605	4.03215	3.23972	1.81617
10	1.15E-08	9.16E-02	4.35E-02	4.42E-02	2.11E-02
11	0	0	0	0	1.54E-09
12	3.19987	8.65935	6.01094	6.14313	1.20528
13	1.87848	11.0887	5.78544	5.63428	1.98031
14	0	1.24E-01	0	3.06E-02	4.21E-02
15	323.745	971.289	726.682	700.799	147.735
16	4.92E-01	1.26998	9.45E-01	9.30E-01	1.75E-01
17	10.1224	10.1224	10.1224	10.1224	1.26E-14
18	14.2817	30.1423	23.7402	23.677	2.87485
19	3.00E-01	5.09E-01	3.77E-01	3.84E-01	4.78E-02
20	1.62304	3.34076	2.18941	2.24119	3.87E-01
21	200	400.194	400.194	392.343	39.2459
22	5.953E-01	27.1744	11.6738	12.7597	5.3465
23	337.038	1224.32	736.82	714.077	197.034
24	107.524	207.33	200	196.213	19.7732
25	200	204.54	200	200.268	1.07861
26	103.52	200.017	200.017	158.229	46.712
27	300	400	300	301.961	14.0028
28	100	300	300	292.157	39.2078

Table 8 Basic Characteristics of function error, b6e6rl, $D = 30$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	0
2	16174.1	175651	57806.1	74639.9	43437.4
3	1.10E-02	588497	21.2778	13494.7	82336.7
4	8.08E-04	3.42E-01	2.57E-02	5.18E-02	6.91E-02
5	0	0	0	0	0
6	2.72E-02	26.4074	6.14E-01	2.69317	7.00534
7	4.71078	49.9142	24.2584	25.6147	11.019
8	20.7806	21.0382	20.9459	20.9423	5.15E-02
9	25.5287	30.9064	29.1343	28.7904	1.44861
10	0	4.67E-02	2.46E-02	2.25E-02	1.29E-02
11	0	0	0	0	0
12	59.2826	114.541	87.5171	86.6021	11.3868
13	84.8923	154.892	118.814	117.396	16.7668
14	0	6.25E-02	2.08E-02	2.37E-02	1.82E-02
15	3909.92	5448.6	4700.77	4656.8	307.573
16	4.07E-01	2.43137	1.981	1.88839	4.22E-01
17	30.4337	30.4338	30.4337	30.4337	2.72E-05
18	136.394	208.337	177.157	176.758	13.9437
19	1.63763	2.03302	1.86691	1.8499	1.13E-01
20	10.6751	12.4114	11.8067	11.8146	3.24E-01
21	200	443.544	300	291.998	83.926
22	31.5253	150.216	124.172	123.035	21.1747
23	4098.14	5564.91	4977.98	4920.14	376.015
24	218.451	281.427	258.063	256.869	15.1052
25	234.863	295.799	282.212	274.477	18.3494
26	200.002	372.637	200.007	206.666	33.2902
27	930.157	1081.53	1021.94	1017.8	38.0067
28	300	300	300	300	0

Table 9 Basic Characteristics of function error, JADE, $D = 30$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	0
2	761.617	34624.7	5211.6	7916.56	6901.76
3	0	9.20E+06	1.80499	564773.1	1.91E+06
4	0	37183.5	1.90E-06	3255.48	9176.52
5	0	0	0	0	3.70E-14
6	0	26.4074	0	2.07117	7.17025
7	4.56E-11	18.3384	2.07641	3.26158	3.71848
8	20.6911	21.0282	20.9337	20.9277	6.30E-02
9	21.4688	29.4765	26.9748	26.7183	1.64676
10	0	1.40E-01	3.69E-02	4.29E-02	2.74E-12
11	0	0	0	0	0
12	12.9951	31.1811	23.4249	23.6092	3.88487
13	17.0193	72.4851	51.615	48.6136	13.1959
14	0	8.33E-02	2.08E-02	2.86E-02	2.46E-02
15	2406.77	3783.8	3288.82	3274.999	309.786
16	1.30E-01	2.73021	1.80939	1.71086	6.20E-01
17	30.4337	30.4337	30.4337	30.4337	2.51E-14
18	61.7734	87.8341	77.0715	76.3183	6.18966
19	1.17009	1.66714	1.43957	1.44109	1.24E-01
20	8.34224	13.2213	10.1325	10.1618	7.84E-01
21	200	443.544	300	312.713	64.2801
22	6.98227	136.709	105.853	88.3946	36.2618
23	2626.44	4537.7	3513.19	3502.51	403.123
24	200.568	264.632	207.894	211.072	10.6093
25	235.557	291.657	281.352	276.363	13.4536
26	200	354.232	200.001	213.461	41.203
27	309.479	1033.09	743.386	688.174	219.546
28	300	1306.51	300	319.736	140.94

Table 10 Basic Characteristics of function error, b6e6pbest, $D = 30$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	3.18E-14
2	2249.91	54497.4	15039	16644.35	11008.9
3	6.44E-08	3.09E+06	2210.58	165954	491649
4	0	1.44E-05	2E-07	1.14E-06	2.54E-06
5	0	0	0	0	0
6	0	26.4074	0	2.58896	7.93085
7	5.73E-01	36.9534	6.05583	9.47878	8.85171
8	20.7537	21.0194	20.9484	20.9315	5.98E-02
9	22.9838	31.6138	28.1173	27.923	1.98579
10	0	1.48E-01	3.69E-02	3.45E-02	2.28E-02
11	0	0	0	0	1.54E-14
12	30.3004	62.5271	45.6549	46.4292	7.91358
13	49.9074	114.544	82.6618	81.87181	15.0246
14	4.20E-01	2.71392	1.13911	1.24838	6.19E-01
15	3584.11	4928.12	4225.56	4223.14	300.524
16	9.58E-01	2.25823	1.78226	1.7925	2.721E-01
17	30.4337	30.4338	30.4337	30.4337	1.40E-05
18	87.7383	137.811	120.229	117.283	11.87165
19	1.26489	1.79963	1.5872	1.58811	1.19E-01
20	9.61117	11.7956	10.9697	10.8991	4.56E-01
21	200	443.544	300	299.841	79.4283
22	26.1122	152.658	119.885	119.606	26.8395
23	3667.03	5451.67	4385.2	4449.63	418.725
24	201.35	267.758	210.287	213.571	12.9728
25	202.194	287.899	246.684	250.746	14.6862
26	200	323.27	200.001	221.84	44.7631
27	306.566	1012.59	416.414	488.991	204.007
28	300	300	300	300	0

Table 11 Basic Characteristics of function error, b3e3j6-F05F08, $D = 30$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	8.21E-10
2	2340.23	80295	21657.1	25671.9	15445.4
3	2.21E-06	36729.6	7.45E-03	742.942	5141.16
4	4.02E-07	5.57E-04	4.53E-05	8.69E-05	1.21E-04
5	0	0	0	0	6.96E-10
6	0	26.4074	0	5.69571	10.9694
7	5.87E-01	29.6898	4.60281	6.75794	6.2968
8	20.796	21.0417	20.9436	20.9446	5.22E-02
9	24.2353	31.8769	28.564	28.4967	1.4852
10	0	5.17E-02	1.48E-02	1.82E-02	1.19E-02
11	0	0	0	0	9.31E-10
12	54.7313	105.007	77.0414	76.4243	11.2371
13	58.5736	129.195	102.306	98.8592	16.5599
14	0	1.04E-01	4.16E-02	4.45E-02	3.03E-02
15	3780.15	5227.68	4610.19	4596.19	325.023
16	1.35655	2.49484	1.96029	1.96159	2.70E-01
17	30.4337	30.4338	30.4337	30.4337	3.25E-05
18	112.661	193.56	161.638	160.769	14.0341
19	1.57246	1.96792	1.82256	1.8129	9.33E-02
20	10.4852	12.1487	11.5357	11.5241	3.30E-01
21	200	443.544	300	289.183	81.1014
22	38.2279	146.836	116.805	118.963	18.8069
23	3904.54	5492.62	4865.36	4805.72	353.329
24	200.596	278.556	207.09	213.686	16.7666
25	232.459	292.798	247.495	255.291	18.8019
26	200	369.029	200.003	217.159	48.1377
27	317.889	1098.11	851.357	770.104	232.374
28	300	300	300	300	0

Table 12 Basic Characteristics of function error, b3e3j6-F05, $D = 30$.

F	Best	Worst	Median	Mean	Std
1	0	0	0	0	7.09E-10
2	10484.3	426505	42174.7	62585.87	68067.6
3	1.53E-05	5.77E+07	206652	2.32E+06	8.47E+06
4	3.45E-03	9.05E-01	3.42E-02	8.84E-02	1.60E-01
5	0	0	0	0	5.47E-10
6	0	26.4074	9.21941	8.83969	7.04633
7	6.41E-02	24.4607	5.0392	7.28517	6.46305
8	20.8224	21.0271	20.9569	20.9507	4.82E-02
9	24.551	30.8601	28.1414	27.88154	1.4936
10	0	1.63E-01	3.70E-02	4.46E-02	3.15E-02
11	0	0	0	0	8.18E-10
12	24.1928	68.6998	43.2775	44.793	10.5106
13	28.5002	99.0808	69.9944	70.5154	15.1454
14	0	1.46E-01	6.25E-02	5.84E-02	3.23E-02
15	3394.96	4928.23	4153.28	4156.85	352.415
16	8.52E-01	2.29457	1.82908	1.8329	2.68E-01
17	30.4337	30.4338	30.4337	30.4337	4.60E-05
18	98.9106	138.029	117.858	116.917	9.08682
19	1.30075	1.85484	1.53537	1.55755	1.24E-01
20	9.84115	11.7368	11.08	10.9825	4.54E-01
21	200	443.544	300	300.094	69.8904
22	18.9138	139.54	113.979	109.86	24.5708
23	2824.29	5166.48	4522.45	4437.44	444.427
24	201.107	244.428	212.776	213.524	8.64678
25	238.777	270.243	246.826	247.716	5.87706
26	200.001	361.121	200.002	216.924	43.7073
27	304.818	969.763	408.779	507.766	209.258
28	300	300	300	300	0