2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Differential Evolution with Rotation-Invariant Mutation and Competing-Strategies Adaptation

Petr Bujok
Department of Computer Science,
University of Ostrava,
30. dubna 22, 701 03 Ostrava,
Czech Republic
Email: petr.bujok@osu.cz

Josef Tvrdík
Department of Computer Science,
Centre of Excellence IT4Innovations,
Institute for Research and
Applications of Fuzzy Modeling,
University of Ostrava,
30. dubna 22, 701 03 Ostrava,
Czech Republic
Email: josef.tvrdik@osu.cz

Radka Poláková
Centre of Excellence IT4Innovations,
Institute for Research and
Applications of Fuzzy Modeling,
University of Ostrava,
30. dubna 22, 701 03 Ostrava,
Czech Republic
Email: radka.polakova@osu.cz

***Abstract*—A new variant of the adaptive differential evolution algorithm was proposed and tested experimentally on the CEC 2014 test suite. In the new variant, the adaptation is based on the competition of several strategies. A part of strategies in the pool uses the rotation-invariant current-to-pbest mutation in the novel algorithm. The aim of the experimental comparison was to find whether the presence of the rotation-invariant strategy is able to improve the efficiency of the differential evolution algorithm, especially in problems with rotated objective functions. The results of the experiments showed that the new variant performed well in a few of the test problems, while no apparent benefit was observed in the majority of the benchmark problems.**

I. Introduction

Differential evolution (DE) proposed in [1] is a population-based optimization algorithm for single-objective problems with a real-valued objective function. The possible solutions in DE are represented as vectors with real-number components, $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$, 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 $\boldsymbol{x}^*$ satisfying condition $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \Omega$.

The population of the size N is developed step-by-step from a generation P to a generation Q. The evolutionary operators, i.e. mutation, crossover, and selection are applied in the development of generation Q. The DE algorithm is shown in pseudo-code in Algorithm 1. The new trial point is created from a mutant point $\boldsymbol{u}$ generated by using a kind of mutation and from the current point of the population by the application of the crossover. A better point from the couple of $\boldsymbol{x}_i$, $\boldsymbol{y}$, based on the value of the objective function, is selected for the new generation Q.

The DE algorithm has been studied intensively in recent times. Comprehensive summary of advanced results in DE research is available in [2] 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 without manual tuning of DE control parameters.

Algorithm 1 Differential evolution algorithm

```
initialize population P = {x_1, x_2, ..., x_N}
while stopping condition not reached do
  for i = 1, 2, ..., N do
    create a new trial vector y
    compute f(y)
    if f(y) ≤ f(x_i) then
      insert y into Q
    else
      insert x_i into Q
    end if
  end for
  P ← Q
end while
```

The remaining part of the paper is organized in the following manner. Adaptive variants of DE from which the new algorithm was inspired are described in Section II. The novel DE algorithm is presented in Section III. Settings of experiments are given in Section IV. The results of the novel algorithm on CEC 2014 problems are depicted in Section V and some conclusions are made in Section VI.

II. Adaptive Variants of Differential Evolution

There are many adaptive DE variants that appeared in literature during last decade. They differ in adaptive mechanisms and/or in combinations of DE strategies applied. Among published adaptive DE variants there are four commonly considered as the state-of-the-art, namely jDE [4], SADE [5], JADE [6], and EPSDE [7]. Several new adaptive variants of DE were introduced on CEC 2013, e.g. [8]–[12].

For our novel algorithm, we borrow concepts from JADE (rotation-invariant current-to-pbest mutation, archive) and the adaptive mechanism of competing strategies from [13], [14]. These algorithms are described below.

A. JADE

JADE variant of adaptive differential evolution [6] 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 $\boldsymbol{u}$ is generated in the following manner:

$$\boldsymbol{u} = \boldsymbol{x}_i + F\,(\boldsymbol{x}_{\text{pbest}} - \boldsymbol{x}_i) + F\,(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}), \tag{1}$$

where $\boldsymbol{x}_{\text{pbest}}$ is randomly chosen from $100\,p\,\%$ best individuals with input parameter $p = 0.05$ recommended in [6]. The vector $\boldsymbol{x}_{r_1}$ is randomly selected from P $(r_1 \neq i)$, $\boldsymbol{x}_{r_2}$ is randomly selected from the union $P \bigcup A$ $(r_2 \neq i \neq r_1)$ 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 $\boldsymbol{u}$ and $\boldsymbol{x}_i$ using the binomial crossover. *CR* and F are independently generated for each individual $\boldsymbol{x}_i$, *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 [6] for details of μ_{CR} and μ_{F} adaptation.

B. Competitive DE

Competitive DE uses H strategies with their control-parameter values held in the pool. Any of H strategies can be chosen to create a new trial point $\boldsymbol{y}$. 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

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

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 $\delta,\ \ \delta > 0$.

We use a variant of competitive DE that appeared well-performing and robust in different benchmark tests [15]. 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 $F = 0.8$.

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

where the point $\boldsymbol{r}_1^x$ is tournament best among $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$, i.e. $f(\boldsymbol{r}_1^x) \leq f(\boldsymbol{r}_j^x),\ j = 2, 3$, as proposed in [16].

Mutation can cause that a mutant point $\boldsymbol{u}$ 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 $u_j \leftarrow 2 \times a_j - u_j$ or $v_j \leftarrow 2 \times b_j - u_j$ for the violated component.

The binomial crossover uses three different values of *CR*, $CR \in \{0, 0.5, 1\}$. The values of *CR* for exponential crossover are evaluated from the polynom equation [17]

$$CR^D - \ D\ p_m\ CR + \ D\ p_m - 1 = 0. \tag{4}$$

Three values of mutation probability p_m are set up equidistantly in the interval $(1/D, 1)$. The values of mutation probability and the corresponding values of *CR* applied to the problems of $D = 10$, 30, 50, and 100 are shown in Table I.

TABLE I. VALUES OF MUTATION PROBABILITY AND THE CORRESPONDING VALUES OF *CR* FOR EXPONENTIAL CROSSOVER

	$D = 10$		$D = 30$	
i	p_i	CR_i	p_i	CR_i
1	0.3250	0.7011	0.2750	0.8815
2	0.5500	0.8571	0.5167	0.9488
3	0.7750	0.9418	0.7583	0.9801
	$D = 50$		$D = 100$	
i	p_i	CR_i	p_i	CR_i
1	0.2650	0.9262	0.2525	0.9611
2	0.5100	0.9688	0.4950	0.9837
3	0.7550	0.9880	0.7475	0.9938

III. Novel Algorithm Combining Competitive Adaptation and JADE

Seven adaptive DE variants [4]–[7], [15], [18] were compared experimentally on six standard benchmark functions at three levels of dimension in [19]. It was found that JADE [6] and *b6e6rl* [15] 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. Four adaptive DE variants, namely JADE, b6e6rl, EPSDE, and CoDE were also compared experimentally on CEC 2013 benchmark functions in [20], where JADE appeared the most efficient, followed by b6e6rl and EPSDE.

It was found in [21] that the adaptive DE b6e6rl variant based on the competition of strategies performs well on the problems, where the objective function is not rotated, whilst the performance in the problems with rotated functions is worse. In this paper, a novel variant of the competitive DE combining two adaptive approaches is proposed. The adaptive DE variant derived from b6e6rl, where the randrl mutation is replaced by the current-to-pbest mutation, was tested on the CEC2013 problem suite [22]. The new algorithm outperformed the parent JADE and b6e6rl only in a few test problems [23].

The adaptive DE algorithm newly proposed for this paper (denoted *b3e3pbest* hereafter) uses the competitive adaptive mechanism described in subsection II-B but the pool of competing strategies is redesigned. Six strategies using randrl mutation are taken from the b6e6rl (all the strategies with $F = 0.8$) and one strategy uses the current-to-pbest mutation from JADE [23] with $F = 0.5$. An archive from JADE containing the old best solutions is also applied. Because of the fact that the current-to-pbest mutation includes arithmetic crossover, no other crossover occurs in this strategy. Such strategy gives the same result as the binomial or exponential crossover with $CR = 1$. It is expected that application of the rotation-invariant current-to-pbest mutation can help in the solution of rotated functions. Thus, seven different DE strategies compete in the search of the global minimum. The new algorithm is shown in pseudo-code in Algorithm 2.

Algorithm 2 Competitive DE algorithm

```
initialize population P = {x_1, x_2, ..., x_N}
initialize empty archive A of the size N
initialize probabilities of strategies
while stopping condition not reached do
  for i = 1, 2, ..., N do
    choose a strategy by a roulette selection
    create a new trial vector y
    compute f(y)
    if f(y) ≤ f(x_i) then
      insert y into Q
      insert x_i into A
      update the value of a probability of used strategy
    else
      insert x_i into Q
    end if
  end for
  P ← Q
end while
```

First, the population P of size N, randomly uniformly distributed in the area of the possible solutions, is initialized. Along with, the empty archive A of the size N for the storage of old solutions is also initialized. When the new trial point is inserted into next generation Q, the old solution $\boldsymbol{x}_i$ is stored in the archive A. If the archive is full, a randomly selected point in the A is replaced by the current $\boldsymbol{x}_i$.

The other control parameters are set up to the recommended values, i.e. $\delta = 1/(5 \times 7) = 0.0286$, $n_0 = 2$ and the control parameter for current-to-pbest mutation $p = 0.05$.

IV. Experiments

The new test suite of 30 functions was proposed for the special session on Real-Parameter Numerical Optimization, a part of Congress on Evolutionary Computation (CEC) 2014. This session is intended as a competition of optimization algorithms. The functions are described in report [24], including the experimental settings required for the competition. The source code of the functions is also available on 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 benchmarks required for publishing new single-objective optimization algorithms.

The algorithm is implemented in Matlab 2010a and this environment was also used for experiments. All computations were carried out on a standard PC with Windows 7, Intel(R) Core(TM)2 CPU 6320, 1.86GH 1.87GH, 2GB RAM.

Experimental setting follows the requirements given in the report [24], where 30 minimization problems are also defined.

TABLE II. Results of function errors for $D = 10$.

Func.	Best	Worst	Median	Mean	Std
1	0	0	0	0	1.60E-09
2	0	0	0	0	1.74E-09
3	0	0	0	0	1.38E-09
4	0	34.7803	34.7803	27.53369	13.9856
5	8.13405	20.1192	20.0739	19.20715	2.730173
6	0	0.003672	0	7.31E-05	0.000514
7	0	0.08993	0.047343	0.04605	0.020512
8	0	0	0	0	1.93E-09
9	2.63731	9.82905	6.31719	6.310018	1.397936
10	0	0	0	0.001225	0.008745
11	95.1019	486.853	315.549	306.0512	115.8411
12	0.197947	0.483994	0.365873	0.363756	0.069213
13	0.050217	0.195818	0.144352	0.139047	0.03014
14	0.062985	0.196257	0.115715	0.116497	0.028335
15	0.568242	1.26531	0.910707	0.928939	0.14502
16	1.51143	2.45688	2.14802	2.140389	0.231109
17	0	129.611	0.416286	4.840349	19.25361
18	0	9.95E-01	0	0.019509	1.39E-01
19	0.058644	0.682468	0.277017	0.281704	0.116528
20	0.042432	0.389328	0.19254	0.206819	0.080309
21	0.043387	0.94029	0.368279	0.369574	0.236436
22	1.44E-06	0.62436	0.031724	0.139596	0.166315
23	329.457	329.457	329.457	329.457	1.72E-13
24	107.039	117.845	112.596	112.2336	2.097979
25	101.729	201.374	119.496	145.2207	39.8456
26	100.08	100.203	100.141	100.1434	0.028952
27	0.78441	400.332	2.00649	41.03604	119.5913
28	384.913	386.547	384.913	385.1086	0.392145
29	220.358	225.054	220.575	221.0688	0.926018
30	308.549	628.744	314.121	365.6103	100.0853

The source code of the functions in C was downloaded from the web page given in [24] 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 four levels of dimension, $D = 10, 30, 50, 100$, with 51 times repeated runs per each test function. The run stops if the prescribed value of MaxFES $= D \times 10^4$ is reached or if the minimum function error in the population is less than 1×10^{-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 1×10^{-8} are treated as zero in further processing.

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

V. Results

The basic characteristics of the experiments are summarized in the Tables II, III, IV, and V. The values of the function errors less than 1×10^{-8} are substituted by zeros in all the tables below. It can be observed that the proposed DE algorithm is able to solve some of the problems. For the

TABLE III. RESULTS OF FUNCTION ERRORS FOR $D = 30$.

Func.	Best	Worst	Median	Mean	Std
1	922.131	44959.8	10441.3	12058.55	9099.398
2	0	0	0	0	6.27E-10
3	0	0	0	0	5.30E-10
4	0	63.4007	0	3.729453	1.51E+01
5	20.2405	20.3834	20.3277	20.323	0.029026
6	9.72593	16.3713	14.2816	14.24751	1.195089
7	0	0.017241	0	0.001353	3.63E-03
8	0	0	0	0	8.91E-10
9	32.456	60.4819	43.8116	45.12717	6.108515
10	0	0.041639	0	0.002449	7.95E-03
11	1778.72	2834.78	2430.44	2420.86	240.3957
12	0.331581	0.531896	0.458726	0.451375	0.049384
13	0.173057	0.42036	0.271947	0.280898	0.054712
14	0.134383	0.299962	0.214213	0.21095	0.032145
15	4.44552	6.93575	5.69338	5.663361	0.617392
16	8.72549	10.5683	10.0337	9.940366	0.379194
17	401.458	1879.48	1080.64	1062.373	320.2895
18	19.8991	153.221	69.6468	74.19489	31.84657
19	2.55363	7.10202	4.63983	4.790698	0.946259
20	8.18754	144.134	18.6493	28.23484	28.9921
21	28.0778	756.986	299.971	319.472	163.3768
22	22.9186	443	131.636	142.3191	94.15324
23	315.244	315.244	315.244	315.244	2.30E-13
24	219.425	237.241	224.089	225.313	4.401665
25	202.558	204.895	203.689	203.6973	0.637062
26	100.153	100.357	100.272	100.2693	0.043459
27	300	403.495	306.275	320.7072	30.69734
28	673.624	917.18	736.105	746.0125	47.60603
29	472.509	862.216	755.361	756.997	52.2784
30	2048.58	4526.12	2786.96	2844.521	458.6856

TABLE IV. RESULTS OF FUNCTION ERRORS FOR $D = 50$.

Func.	Best	Worst	Median	Mean	Std
1	13179.3	369377	85708.2	103490.9	72986.81
2	0	0.001191	3.18E-06	6.43E-05	0.000199
3	4.01E-05	0.828042	0.011123	0.042915	0.120233
4	4.47E-08	98.3971	2.48028	16.20268	29.98973
5	20.3627	20.4718	20.4389	20.43241	0.025996
6	16.9153	33.173	29.3636	29.13724	2.562597
7	0	0.049176	0	0.005795	0.008787
8	0	0	0	0	5.14E-10
9	70.0044	129.68	102.874	103.968	13.50471
10	0	0.012492	0	0	0.004588
11	4418.35	5765.96	5341.12	5297.332	305.0039
12	0.368542	0.570135	0.478765	0.473602	0.039048
13	0.311991	0.584539	0.388711	0.39763	0.058991
14	0.179417	0.318381	0.251103	0.247495	0.030148
15	11.1216	20.7574	14.1781	14.48363	1.936628
16	17.6459	18.9716	18.4488	18.4054	0.311984
17	1283.27	15911.2	4026.2	4625.437	2482.289
18	71.6774	580.431	205.994	224.9669	87.69431
19	9.34958	80.1077	15.216	17.65322	12.78621
20	111.123	628.52	324.499	346.3786	105.7916
21	1114.21	6897.12	1898.41	2141.006	961.4198
22	151.328	900.828	602.166	591.5764	189.3
23	344.005	344.005	344.005	344.005	2.87E-13
24	280.394	302.099	289.082	288.8625	4.406833
25	205.312	229.418	209.808	211.6928	5.636189
26	100.252	200.04	100.361	104.2885	19.53723
27	399.228	677.869	563.471	548.5508	73.36652
28	876.661	1480.57	1019.86	1074.298	166.5429
29	773.176	1177.81	885.483	896.2533	78.59248
30	11947.7	19905.7	14147.8	14156.78	1127.687

problems defined by the composed functions from F23 to F30, new algorithm is unable to find an acceptable solution for all levels of the dimension. Based on the medians of the error, the algorithm was successful in 7 problems for $D = 10$, in 6 problems for $D = 30$, in 3 problems for $D = 50$, and in 3 problems for $D = 100$.

In spite of the fact that the rotation-invariant mutation is used, the performance of the algorithm on rotated benchmark functions is worse at all the levels of dimension. The contrast in the algorithm performance is apparently visible when we compare the pairs of functions $\{8,\ 9\}$ and $\{10,\ 11\}$ which differ only by rotation, the second function of the pair is the same as the first but rotated. Convergence plots for these pairs of functions and $D = 30$ are depicted in Figures 1 and 2. It is apparent that the convergence of the algorithm on rotated functions is much smaller compared to the unrotated ones.

Frequency of the exploitation of different DE strategies from the pool is shown in Figures 3 and 4. The box plots are evaluated from 51 repeated runs of the algorithm on each test problem. The strategies are labeled from s1 to s7 on the horizontal axis. The strategy with the rotation-invariant current-to-pbest mutation is denoted "s1". The strategies using the randrl rotation and the binary crossover are labeled "s2",

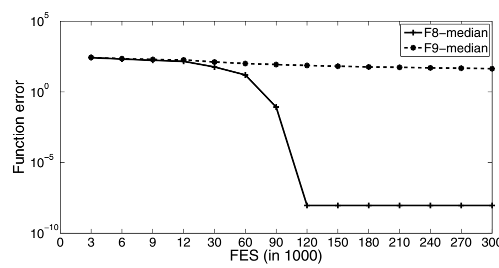

Fig. 1. Convergence plot of F8 and F9 test problems, $D = 30$

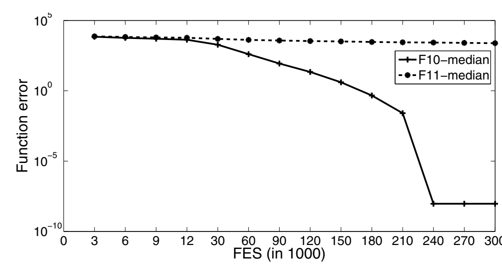

Fig. 2. Convergence plot of F10 and F11 test problems, $D = 30$

TABLE V. RESULTS OF FUNCTION ERRORS FOR $D = 100$.

Func.	Best	Worst	Median	Mean	Std
1	162090	1292790	435189	434103.3	206921
2	14.9169	106649	7254.73	15705.76	20889.35
3	3.23437	633.211	41.2978	91.28251	142.6779
4	75.9957	255.187	148.681	142.8395	39.25807
5	20.6245	20.7139	20.678	20.67425	0.021551
6	46.0904	81.5387	74.4628	69.4176	10.46363
7	0	0.061428	0	0.006315	1.23E-02
8	0	0	0	0	1.64E-10
9	192.153	383.95	302.119	310.6192	40.81764
10	0	0.018738	0	0.003674	4.53E-03
11	8929.85	14420.4	13275.6	12900.58	1365.915
12	0.250135	0.730922	0.661643	0.644243	0.074876
13	0.370024	0.620064	0.488726	0.489832	0.060413
14	0.217761	0.367661	0.301252	0.298722	0.030627
15	30.2445	81.7891	57.4855	56.80116	11.66319
16	39.0653	41.0041	40.4451	40.375	0.462606
17	14680.2	119667	43703	45136.86	18050.84
18	318.998	8782.69	1369.1	2345.985	2130.265
19	27.8039	130.083	101.452	86.53414	31.6961
20	404.092	2199.75	867.792	950.2607	341.9224
21	6195.83	78223.1	22330.7	26165.88	16559.49
22	663.396	2364.72	1595.58	1603.398	339.7521
23	3.48E+02	3.48E+02	348.235	348.235	2.30E-13
24	398.055	435.846	414.911	416.0454	8.062885
25	200	290.366	266.663	263.6285	13.91351
26	200.042	200.118	200.094	200.095	0.012032
27	1226.77	2135.95	1772.25	1771.73	168.351
28	1520.92	3385.76	2341.1	2254.01	378.891
29	714.162	2123.07	904.882	1134.9	432.455
30	7247.86	15112	10220.2	10849.7	2008.2

"s3", and "s4", the strategies with exponential crossover have labels "s5", "s6", and "s7" and are orderer with respect to the values of mutation probability in ascending way. In spite of the expectation, the rotation-invariant current-to-pbest mutation is more frequently used in not-rotated function than in rotated one in both the pairs of the functions. A most prevalently used DE strategy in all the problems is s2 strategy, where only one element of the trial vector comes out from the mutant vector, while the other strategies are rarely successful (their frequency is mostly much less than 10 %). Here is a field for further research.

The complexity of the algorithm evaluated according to the requirement specified in the report [24] is presented in Table VI. The values of time are given in seconds.

TABLE VI. COMPUTATIONAL COMPLEXITY.

	$T0$ (s)	$T1$ (s)	$\hat{T}2$ (s)	$(\hat{T}2 - T1)/T0$
D=10	0.2687	3.55	33.93	113.06
D=30	0.2687	5.47	36.42	115.18
D=50	0.2687	8.30	42.65	127.84
D=100	0.2687	21.76	62.50	151.62

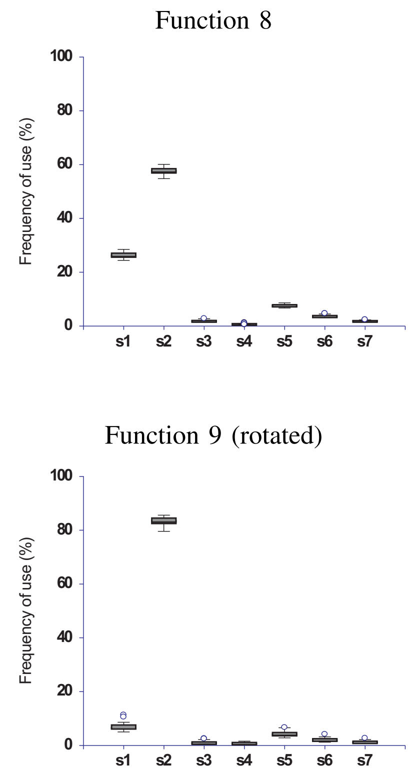

Fig. 3. Frequency of strategies used in the solution of F8 and F9 test problems, $D = 30$

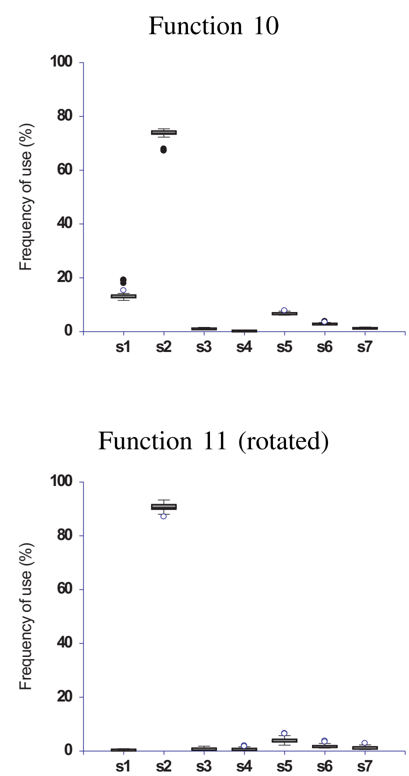

Fig. 4. Frequency of strategies used in the solution of F10 and F11 test problems, $D = 30$

VI. Conclusion

The experimental comparison shows that the novel *b3e3-pbest* variant of competitive DE algorithm is able to solve several problems of CEC 2014 suite but it cannot be considered completely satisfactory. The algorithm does not perform well on the most of the problems with rotated objective functions or on composition functions. The inclusion of the rotation-invariant current-to-pbest mutation into the pool of competing DE strategies does not cause sufficient enhancement of the algorithm performance.

Thus, the proposal of innovated DE variant increasing the efficiency of DE on rotated or composition functions remains the challenge for further research.

Acknowledgment

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

References

[1] R. Storn and K. V. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, pp. 341–359, 1997.

[2] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, pp. 61–106, 2010.

[3] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 27–54, 2011.

[4] J. Brest, S. Greiner, B. Boškovič, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 646–657, 2006.

[5] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[6] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 945–958, 2009.

[7] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, pp. 1679–1696, 2011.

[8] R. Tanabe and A. Fukunaga, "Evaluating the performance of shade on cec 2013 benchmark problems," in *IEEE Congress on Evolutionary Computation 2013 Proceedings*, 2013, pp. 1952–1959.

[9] F. Caraffini, F. Neri, J. Cheng, G. Zhang, L. Picinali, G. Iacca, and E. Mininno, "Super-fit multicriteria adaptive differential evolution," in *IEEE Congress on Evolutionary Computation 2013 Proceedings*, 2013, pp. 1678–1685.

[10] S. Biswas, S. Kundu, S. Das, and A. V. Vasilakos, "Teaching and learning based differential evoltuion with self adaptation for real parameter optimization," in *IEEE Congress on Evolutionary Computation 2013 Proceedings*, 2013, pp. 1115–1122.

[11] I. Poikolainen and F. Neri, "Differential evolution with concurrent fitness based local search," in *IEEE Congress on Evolutionary Computation 2013 Proceedings*, 2013, pp. 384–391.

[12] A. Zamuda, J. Brest, and E. Mezura-Montes, "Structured population size reduction differential evolution with multiple mutation strategies on cec 2013 real parameter optimization," in *IEEE Congress on Evolutionary Computation 2013 Proceedings*, 2013, pp. 1925–1931.

[13] J. Tvrdík, "Competitive differential evolution," in *MENDEL 2006, 12th International Conference on Soft Computing*, R. Matoušek and P. Ošmera, Eds. Brno: University of Technology, 2006, pp. 7–12.

[14] J. Tvrdík, "Adaptation in differential evolution: A numerical comparison," *Applied Soft Computing*, vol. 9, no. 3, pp. 1149–1155, 2009.

[15] J. Tvrdík, "Self-adaptive variants of differential evolution with exponential crossover," *Analele of West University Timisoara, Series Mathematics-Informatics*, vol. 47, pp. 151–168, 2009. [Online]. Available: http://www1.osu.cz/˜tvrdik/

[16] P. Kaelo and M. M. Ali, "A numerical study of some modified differential evolution algorithms," *European J. Operational Research*, vol. 169, pp. 1176–1184, 2006.

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

[18] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 55–66, 2011.

[19] J. Tvrdík, R. Poláková, J. Veselský, and P. Bujok, "Adaptive variants of differential evolution: Towards control-parameter-free optimizers," in *Handbook of Optimization - From Classical to Modern Approach*, ser. Intellingent Systems Reference Library, vol. 38, I. Zelinka, V. Snšel, and A. Abraham, Eds. Berlin Heidelberg: Springer-Verlag, 2012, pp. 423–449.

[20] J. Tvrdík, P. Bujok, and R. Poláková, "A comparison of adaptive differential evolution algorithms on cec 2013 benchmark problems," in *19th International Conference on Soft Computing MENDEL 2013*. Brno: Brno University of Technology, 2013, pp. 123 – 128.

[21] J. Tvrdík and R. Polakova, "Competitive Differential Evolution Applied to CEC 2013 Problems," in *IEEE Congress on Evolutionary Computation 2013*. IEEE Computational Intelligence Society, 2013, pp. 1651–1657.

[22] J. J. Liang, B. Qu, P. N. Suganthan, and A. G. Hernandez-Diaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Nanyang Technological University, Singapore, 2013. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/

[23] P. Bujok and J. Tvrdík, "A new variant of competitive differential evolution algorithm," in *INFORMATICS'2013 - International Scientific Conference on Informatics*. Košice: FEEI TU of Košice, 2013, pp. 304–308.

[24] J. J. Liang, B. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," Nanyang Technological University, Singapore, 2013. [Online]. Available: http://www.ntu.edu.sg/home/epnsugan/