A COMBINED APPROACH TO ADAPTIVE DIFFERENTIAL EVOLUTION

Radka Poláková, Josef Tvrdík[*]

Abstract: The paper deals with the adaptive mechanisms in differential evolution (DE) algorithm. DE is a simple and effective stochastic algorithm frequently used in solving the real-world global optimization problems. The efficiency of the algorithm is sensitive to setting its control parameters. Several adaptive approaches have appeared recently in order to avoid control-parameter tuning. A new adaptive variant of differential evolution is proposed in this study. It is based on a combination of two adaptive approaches published before. The new algorithm was tested on the well-known set of benchmark problems developed for the special session of CEC2005 at four levels of population size and its performance was compared with the adaptive variants that were applied in the design of the new algorithm. The new adaptive DE variant outperformed the others in several test problems but its efficiency on average was not better.

Key words: *Global optimization, differential evolution, adaption, combined adaptive mechanism, experimental comparison*

Received: July 15, 2012
Revised and accepted: January 7, 2013

1. Introduction

Optimization problems are frequent part of human activities in many fields. We address the single-objective optimization problem defined as follows: for a given objective function $f : S \to \mathcal{R}$, $S \subset \mathcal{R}^d$ we are searching for a point $\boldsymbol{x}^*$ which is called the global minimum point when it satisfies

$$\forall \boldsymbol{x} \in S, f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}). \tag{1}$$

The search space S is closed compact set $S = \prod_{i=1}^{D}[a_i, b_i]; \quad a_i < b_i, \quad i = 1, 2, \ldots, D$. The $f(\boldsymbol{x})$ can be evaluated at any point $\boldsymbol{x} \in S$.

It is well-known that there is no deterministic algorithm solving this problem in polynomial time [1] in general. Standard iterative deterministic algorithms tend

[*]Radka Poláková, Josef Tvrdík
Centre of Excellence IT4Innovations division of University of Ostrava
Institute for Research and Applications of Fuzzy Modeling, E-mail: `radka.polakova@osu.cz`, `josef.tvrdik@osu.cz`

to stop the search in a local minimum nearest to the input starting point. Therefore, stochastic search is widely used for solving the global optimization problem. Differential evolution (DE) is one of such algorithms. DE was proposed by Storn and Price [13, 14] and appeared to be an efficient algorithm in many optimization problems [10].

The aim of this study is to propose a new adaptive mechanism for DE and compare its performance with other well-performing DE variants on hard benchmark problems defined for CEC2005 competition in [15]. This adaptive mechanism is based on a combination of two successful adaptive mechanisms described in literature [2, 16, 17]. The remaining part of the paper is organized as follows: DE algorithm and its control parameters are summarized in Section 2. Two adaptive mechanisms combined in a newly proposed algorithm are described in Sections 3 and 4. The new adaptive variant of DE combining two adaptive mechanisms is proposed in Section 5. Experiments and their results are presented in Section 6 and concluding remarks are made in the last section.

2. Differential Evolution

Differential evolution (DE) is a population based algorithm which was introduced by Storn and Price [14] as a global optimizer for continuous optimization problems with a real-value objective function. DE works with two alternating generations of population, P and Q. The points of population are considered as candidates of solution. At the beginning, the generation P is initialized randomly in the search domain S, $S = \prod_{j=1}^{D}[a_j, b_j],\ a_j < b_j,\ j = 1, 2, \ldots, D$. A point $\boldsymbol{y}$, called the trial point, is computed by mutation and crossover operations for each point $\boldsymbol{x}_i \in P$, $i \in \{1, 2, \ldots, NP\}$, where NP is the size of population. The point $\boldsymbol{y}$ is inserted into new generation Q if $f(\boldsymbol{y}) \le f(\boldsymbol{x}_i)$, otherwise the point $\boldsymbol{x}_i$ enters into Q. After the new generation Q is completed, Q becomes the old generation P and the whole process continues until the stopping condition is satisfied. The basic scheme of DE is shown in a pseudo-code in Algorithm 1.

Algorithm 1 Differential evolution

```
generate an initial population $P = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{NP})$, $\boldsymbol{x}_i \in S$ distributed uniformly
while stopping condition not reached do
  for $i = 1$ to $NP$ do
    generate a trial point $\boldsymbol{y}$
    if $f(\boldsymbol{y}) \le f(\boldsymbol{x}_i)$ then
      insert $\boldsymbol{y}$ into new generation $Q$
    else
      insert $\boldsymbol{x}_i$ into new generation $Q$
    end if
  end for
  $P := Q$
end while
```

The trial point $\boldsymbol{y}$ is generated for each point $\boldsymbol{x}_i$ in the following way. The mutant $\boldsymbol{v}$ is computed by some kind of mutation, and then from mutant $\boldsymbol{v}$ and the current point $\boldsymbol{x}_i$ the trial point $\boldsymbol{y}$ is created by a kind of crossover.

Many kinds of mutation have been proposed [3, 6, 14]. The kinds used in the proposed algorithm are described below. Suppose that $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$ are three mutually distinct points taken randomly from population P, not coinciding with the current $\boldsymbol{x}_i$, $F > 0$ is an input control parameter, and $\mathrm{rand}(0,1)$ is a random number uniformly distributed between 0 and 1. The mutant vector $\boldsymbol{v}$ can be generated as follows:

- rand/1
$$\boldsymbol{v} = \boldsymbol{r}_1 + F\left(\boldsymbol{r}_2 - \boldsymbol{r}_3\right), \tag{2}$$

- randrl/1
$$\boldsymbol{v} = \boldsymbol{r}_1^x + F\left(\boldsymbol{r}_2^x - \boldsymbol{r}_3^x\right), \tag{3}$$
  where the point $\boldsymbol{r}_1^x$ is the best point among $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$, i.e. $\boldsymbol{r}_1^x = \arg\min_{i\in\{1,2,3\}} f(\boldsymbol{r}_i)$, as proposed in [6], and $\boldsymbol{r}_2^x$, $\boldsymbol{r}_3^x$ are remaining points of $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$.

- current-to-rand/1
$$\boldsymbol{y} = \boldsymbol{x}_i + \mathrm{rand}(0,1) \times (\boldsymbol{r}_1 - \boldsymbol{x}_i) + F\left(\boldsymbol{r}_2 - \boldsymbol{r}_3\right). \tag{4}$$
  Note that the current-to-rand/1 mutation generates a trial point $\boldsymbol{y}$ directly, because (4) includes so called arithmetic crossover.

The crossover operator constructs the trial vector $\boldsymbol{y}$ from the current individual $\boldsymbol{x}_i$ and the mutant vector $\boldsymbol{v}$. Two types of crossover were proposed by Storn and Price in [14]. Binomial crossover combines the elements of vector $\boldsymbol{x}_i$ and mutant vector $\boldsymbol{v}$ using the following rule

$$y_j = \begin{cases} v_j & \text{if} \quad U_j \le CR \quad \text{or} \quad j = l \\ x_{ij} & \text{if} \quad U_j > CR \quad \text{and} \quad j \ne l\,, \end{cases} \tag{5}$$

where l is a randomly chosen integer from $\{1, 2, \ldots, D\}$, and $U_1, U_2, \ldots, U_D$ are independent random variables uniformly distributed in $[0,\ 1)$. $CR \in [0, 1]$ is a control parameter influencing the number of elements to be exchanged by crossover. Eq. (5) ensures that $\boldsymbol{y}$ and $\boldsymbol{x}_i$ are different at least in one coordinate, even if $CR = 0$.

In the exponential crossover, the starting position of crossover is also chosen randomly from $1, \ldots, D$, but L consecutive elements (counted in circular manner) are taken from the mutant vector $\boldsymbol{v}$. Probability of replacing the kth element in the sequence $1, 2, \ldots, L,\ \ L \le D$, decreases exponentially with increasing k. L adjacent coordinates are changed in exponential variant while in binomial one the changed coordinates are dispersed randomly over the coordinates $1, 2, \ldots, D$.

Unlike in binomial crossover the relation between the probability of mutation and the CR is nonlinear in the exponential crossover. The deviation from the linearity enlarges with increasing dimension of problem. Probability of mutation (p_m) controls the number of exchanged elements in the crossover, $p_m \times D$ is the expected number of mutant elements used in producing offsprings. Zaharie [19, 20] derived the relation between p_m and CR for exponential crossover. Her result can be rewritten in the form of polynomial equation

$$CR^D - D\, p_m\, CR + D\, p_m - 1 = 0\,. \tag{6}$$

The value of CR for given value of $p_m \in (1/D, 1)$ can be evaluated as the root of the equation (6).

Compared to the other evolutionary algorithms, DE has a very small number of control parameters. When the standard differential evolution algorithm is used to solve a particular optimization problem, the user needs to select a DE strategy, to set up the size of population and the values of F and CR. Using the DE strategy means a combination of mutation and crossover, usually abbreviated by DE/m/n/c, where m stands for a kind of mutation, n for the number of differences of randomly selected points in mutation, and c for the type of crossover. It was found in many studies that the performance of DE is sensitive to the values of control parameters, especially to F and CR, and tuning to the values appropriate for the solved problem can take a lot of time. To avoid this time-consuming process of parameter setting, many sophisticated and adaptive variants of DE have been proposed recently [2, 4, 5, 7, 8, 11, 12, 21].

3. Competitive Differential Evolution

Adaptive mechanism for DE algorithm based on the competition of different DE strategies or F and CR settings was introduced in [16]. Let us have H strategies. By the strategy we mean the DE strategy together with the fix values of mutation parameter F and crossover parameter CR.

Any of H strategies can be chosen for the generation of a new trial point $\boldsymbol{y}$. A strategy is selected randomly with probability q_h, $h = 1, 2, \ldots, H$. At the start, the values of probability are set up uniformly, $q_h = 1/H$, and they are modified according to their success rates in the preceding steps of the search process. The hth strategy is considered successful if it generates such a trial vector $\boldsymbol{y}$ satisfying $f(\boldsymbol{y}) \le f(\boldsymbol{x}_i)$. Probabilities q_h $(h = 1, \ldots, H)$ are evaluated as the relative frequency according to

$$q_h = \frac{n_h + n_0}{\sum_{j=1}^{H}(n_j + n_0)}\,, \tag{7}$$

where n_h is the current count of the hth strategy 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 strategy-choosing process, the current values of q_h are reset to their starting values if any probability q_h decreases below some given limit δ, $\delta > 0$.

4. jDE

This simple and efficient adaptive mechanism for DE algorithm was proposed by Brest et al. [2] and nowadays is considered one of the state-of-the-art adaptive DE algorithms [3]. DE/rand/1/bin is used and the values of F and CR parameters are evolutionary self-adapted. Each point of the population has its own values of parameters F and CR. These values survive when the trial point computed by them is successful, i.e. the trial vector is inserted into next generation.

The values of F and CR are initialized randomly for each point of the population from the uniform distributions, $CR \in [0, 1]$ and $F \in [F_l, F_u]$, F_l and F_u are input parameters of the algorithm. Before computing a new trial point, the values of F and CR are mutated with given probabilities τ_1 and τ_2 by the values from the same distributions as used for their initialization. If the mutation condition happens, the mutated values of F and CR are used in generating a trial vector. If the values of F and CR generate a successful trial point, they are stored in the new generation together with the trial point instead of the current point $\boldsymbol{x}_i$, otherwise the current point together with its old values of F and CR continues in the new generation. Input parameters of *jDE* algorithm in our experiments are set up to $F_l = 0.1$, $F_u = 0.9$, $\tau_1 = 0.1$, and $\tau_2 = 0.1$ as applied in [2].

5. New Adaptive DE Algorithm

The adaptive mechanism of the newly proposed algorithm combines the evolutionary adaptation of F and CR used in *jDE* algorithm [2] with the strategy competition proposed in competitive differential evolution [16]. Three DE strategies compete and the F and CR parameters are adapted during the search process for each strategy separately. In the proposed variant, called *comp3jDE* hereafter, the following different DE strategies are used:

- DE/rand/1/bin using mutation according to (2) together with binomial crossover,
- DE/randrl/1/exp using mutation according to (3) together with exponential crossover,
- DE/current-to-rand/1 using mutation according to (4) without any additional crossover.

This combination of strategies was chosen due to the fact that each of them is supposed to have different performance for different optimization problems [8].

Each point of population has own F and CR parameters for each DE strategy, i.e. five parameters are connected to each point of the population, namely F, CR for DE/rand/1/bin strategy, F and CR for DE/randrl/1/exp strategy, and F for DE/current-to-rand/1 strategy. These three DE strategies compete according to the rules described in Section 3, and parameters of each DE strategy are self-adapted by the mechanism of *jDE* described in Section 4.

6. Experiments and Results

The algorithms were tested on 25 benchmark functions defined for CEC2005 competition and the experiments were carried out in accordance with the suggestions given in [15]. This benchmark is often required by editors when the performance of a new algorithm is demonstrated in a publication. 25 independent runs for each

Functions	ε
f1 – f5	1×10^{-6}
f6 – f16	1×10^{-2}
f17 – f25	1×10^{-1}

Tab. I *Required accuracy ε for the benchmark functions.*

benchmark function and each algorithm were carried out. Each run was stopped if the number of function evaluation $FES = 3 \times 10^5$ was achieved. The function error value (defined as $f(\boldsymbol{x}_{min}) - f(\boldsymbol{x}^*)$, where $\boldsymbol{x}_{min}$ is the solution found in the run and $\boldsymbol{x}^*$ is the global minimum of the benchmark function [15]) was computed in each run.

In order to find the influence of the population size on the performance of the algorithms, *comp3jDE* and *jDE* algorithms were tested at four levels of population size, $NP = 30, 60, 90, 120$. The means and the standard deviations of the function error values are given in Tabs. V and VI in Appendix.

	median				rank			
func.	NP=30	NP=60	NP=90	NP=120	NP=30	NP=60	NP=90	NP=120
f1	0	0	0	0	2.5	2.5	2.5	2.5
f2	0	0	1.50E-06	4.58E-05	1.5	1.5	3	4
f3	1.44E+05	1.24E+05	1.43E+05	1.99E+05	2	2	2	4
f4	2.97E+00	9.74E-03	1.90E-02	8.14E-02	4	2.5	2.5	3
f5	1.49E+03	8.97E+02	3.45E+02	3.87E+02	4	3	1.5	1.5
f6	0	4.79E-01	8.72E+00	1.41E+01	1.5	1.5	3	4
f7	0	1.48E-02	0	0	3.5	3.5	1.5	1.5
f8	2.09E+01	2.09E+01	2.09E+01	2.10E+01	2.5	2.5	2.5	2.5
f9	0	0	0	0	4	2	2	2
f10	3.18E+01	4.21E+01	4.81E+01	5.53E+01	1	2	3	4
f11	2.53E+01	2.65E+01	2.73E+01	2.81E+01	1.5	1.5	3.5	3.5
f12	4.60E+03	3.26E+03	1.47E+04	1.52E+04	1.5	1.5	3.5	3.5
f13	9.93E-01	1.31E+00	1.48E+00	1.63E+00	1	2.5	2.5	4
f14	1.27E+01	1.29E+01	1.30E+01	1.31E+01	1	3	3	4
f15	400	400	400	400	2.5	2.5	2.5	2.5
f16	7.00E+01	6.52E+01	6.75E+01	7.35E+01	3	1	3	3
f17	9.10E+01	1.04E+02	1.20E+02	1.35E+02	1.5	1.5	3.5	3.5
f18	9.07E+02	9.04E+02	9.04E+02	9.04E+02	4	2	2	2
f19	9.07E+02	9.05E+02	9.04E+02	9.04E+02	4	3	1.5	1.5
f20	9.07E+02	9.05E+02	9.04E+02	9.04E+02	4	2	2	2
f21	5.00E+02	5.00E+02	5.00E+02	5.00E+02	2.5	2.5	2.5	2.5
f22	8.81E+02	8.84E+02	8.76E+02	8.67E+02	2.5	2.5	2.5	2.5
f23	5.34E+02	5.34E+02	5.34E+02	5.34E+02	4	2	2	2
f24	200	200	200	200	4	2	2	2
f25	2.13E+02	2.11E+02	2.11E+02	2.10E+02	4	3	1.5	1.5
				average rank	2.70	2.22	2.44	2.76

Tab. II *Comparison of jDE variants using different levels of the population size.*

Comparison of DE variants using different levels of the population size was carried out by one-way analysis of variance, separately for each type of algorithm (*comp3jDE* and *jDE*) and each benchmark function. Due to the violation of normality assumptions in most test problems the non-parametric Kruskal-Wallis method was applied together with non-parametric multiple comparison. The required accuracy of solution is defined for the benchmark functions in [15], see Tab. I. The solutions satisfying the prescribed accuracy are considered equivalent and they were replaced by zero before starting the non-parametric comparison. The medians of variants with various levels of the population size and the rank of variant's performance based on the results of the multiple comparison are presented in Tabs. II and III. If there are two or more variants not differing significantly, they are assigned by their average rank. As we can see in Tab. II, *jDE* algorithm performed best at $NP = 60$, where the average rank is the smallest (2.22), while *comp3jDE* showed the best performance with $NP = 120$, where the average rank is 1.84.

	median				rank			
func.	NP=30	NP=60	NP=90	NP=120	NP=30	NP=60	NP=90	NP=120
f1	0	0	0	0	4	2	2	2
f2	8.76E-02	0	0	0	4	2	2	2
f3	1.65E+06	8.20E+05	5.12E+05	3.91E+05	4	3	1.5	1.5
f4	2.69E+03	2.97E+01	2.76E-02	5.00E-05	4	3	2	1
f5	5.66E+03	3.47E+03	2.54E+03	2.25E+03	4	3	1.5	1.5
f6	8.21E+01	1.18E+01	1.65E+01	1.75E+01	4	2	2	2
f7	1.48E-02	1.23E-02	0	1.97E-02	2.5	2.5	2.5	2.5
f8	2.10E+01	2.09E+01	2.09E+01	2.09E+01	2.5	2.5	2.5	2.5
f9	3.98E+00	0	0	0	4	3	1.5	1.5
f10	9.55E+01	5.47E+01	4.58E+01	3.48E+01	4	2.5	2.5	1
f11	2.14E+01	1.76E+01	1.37E+01	1.25E+01	4	3	1.5	1.5
f12	1.18E+04	5.60E+03	3.52E+03	4.42E+03	4	2	2	2
f13	1.24E+00	1.63E+00	1.86E+00	1.95E+00	1	2.5	2.5	4
f14	1.21E+01	1.24E+01	1.26E+01	1.27E+01	1	2	3.5	3.5
f15	4.17E+02	4.00E+02	4.02E+02	4.00E+02	2.5	2.5	2.5	2.5
f16	1.58E+02	7.45E+01	6.08E+01	5.32E+01	4	3	1.5	1.5
f17	1.17E+02	7.35E+01	6.25E+01	5.44E+01	4	3	1.5	1.5
f18	9.56E+02	9.19E+02	9.15E+02	9.09E+02	4	3	1.5	1.5
f19	9.38E+02	9.20E+02	9.11E+02	9.11E+02	4	3	1.5	1.5
f20	9.46E+02	9.16E+02	9.14E+02	9.08E+02	4	2.5	2.5	1
f21	1.18E+03	5.00E+02	5.00E+02	5.00E+02	4	3	1.5	1.5
f22	9.51E+02	9.22E+02	8.99E+02	9.04E+02	3.5	3.5	1.5	1.5
f23	1.06E+03	5.57E+02	5.40E+02	5.34E+02	4	3	1.5	1.5
f24	200	200	200	200	4	2	2	2
f25	2.28E+02	2.13E+02	2.13E+02	2.12E+02	4	3	1.5	1.5
				average rank	3.56	2.66	1.94	1.84

Tab. III *Comparison of comp3jDE variants using different levels of the population size.*

The best performing *jDE* and *comp3jDE* variants were compared with a competitive DE variant. This DE variant (denoted *b6e6rl*) is described in [18] and it was also successfully tested on CEC2005 benchmark [9]. The comparison was again carried out with the use of non-parametric one-way analysis of variance, separately for each benchmark function. The results of the comparison are given in Tab. IV. According to the average rank, the most successful algorithm was *b6e6rl* followed by *jDE*, and *comp3jDE* was the worst. However, when counting the number of winning positions, *comp3jDE* with three wins was the second best after *b6e6rl* with 5 wins, while standard *jDE* achieved only one win. On other hand, *jDE* was the the worst only twice, while *comp3jDE* was outperformed by the other two algorithms in comparison 10 times. The proposed combination of three strategies in the adaptive mechanism is beneficial for some optimization problems but in spite of expectations it is less robust compared to *jDE* using only one strategy.

	median			rank		
func.	b6e6rl60	jDE60	comp3jDE120	b6e6rl60	jDE60	comp3jDE120
f1	0	0	0	2	2	2
f2	0	0	0	2	2	2
f3	7.69E+04	1.24E+05	3.91E+05	1	2	3
f4	0	9.74E-03	5.00E-05	1	3	2
f5	2.87E+02	8.97E+02	2.25E+03	1	2	3
f6	0	4.79E-01	1.75E+01	1	2	3
f7	0	1.48E-02	1.97E-02	1	2.5	2.5
f8	2.10E+01	2.09E+01	2.09E+01	2	2	2
f9	0	0	0	2	2	2
f10	6.52E+01	4.21E+01	3.48E+01	3	1.5	1.5
f11	2.66E+01	2.65E+01	1.25E+01	2.5	2.5	1
f12	1.63E+04	3.26E+03	4.42E+03	3	1.5	1.5
f13	1.42E+00	1.31E+00	1.95E+00	2	1	3
f14	1.27E+01	1.29E+01	1.27E+01	1.5	3	1.5
f15	400	400	400	1.5	1.5	3
f16	9.51E+01	6.52E+01	5.32E+01	3	2	1
f17	1.38E+02	1.04E+02	5.44E+01	3	2	1
f18	9.05E+02	9.04E+02	9.09E+02	1.5	1.5	3
f19	9.05E+02	9.05E+02	9.11E+02	1.5	1.5	3
f20	9.05E+02	9.05E+02	9.08E+02	1.5	1.5	3
f21	5.00E+02	5.00E+02	5.00E+02	2	2	2
f22	8.87E+02	8.84E+02	9.04E+02	1.5	1.5	3
f23	5.34E+02	5.34E+02	5.34E+02	1.5	1.5	3
f24	200	200	200	2	2	2
f25	2.11E+02	2.11E+02	2.12E+02	2	2	2
			average rank	1.84	1.92	2.24
			# wins	5	1	3

Tab. IV *Comparison of the newly proposed algorithm with two well-performing adaptive DE variants.*

Rates of successful use of DE strategies in *comp3jDE* using $NP = 120$ were monitored in 20 periods of the search process. The average number of success in 25 runs was computed in each period for each strategy. It was observed that the development of the strategy frequencies can be classified into seven different categories:

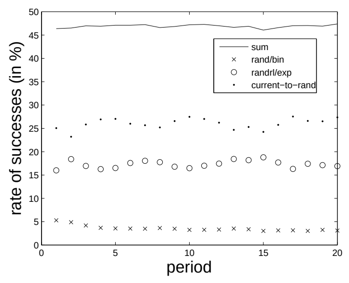
- f2–f7 and f12 – The overall success rate is constant through the search process, DE/current-to-rand/1 was used most frequently and DE/rand/1 least frequently. The development is almost the same as for f6 depicted in Fig. 1a for all those functions.
- f18–f20 – The overall success rate at the beginning is about 1/3 and slightly decreases and then it increases up to about 50 % till the end of the search. The rates of strategies are ordered the same way as in the previous category. The development is similar to f18 depicted in Fig. 1b.
- f1, f21, f24 – At the beginning, the overall success rate is almost 50 % and decreases to zero in the middle of the search, then the population development stagnates either due to the correct solution is found in the case of f1 or population is not able to find any point substantially better. The development is almost the same as for f1 depicted in Fig. 1c.
- f13, f22, f23, f25 – The overall success rate at the beginning is about 1/3 and decreases rapidly to less 5 %. In the case of f13, the DE/randrl/1/exp is the most frequent strategy, while the DE/current-to-rand/1 is the most frequent strategy for the other functions. The development for f23 is depicted in Fig. 1d.
- f10, f11, f15–f17 – The overall success rate at the beginning is less than 5 %, then it increases up to about 1/3 at the end of the search. DE/current-to-rand/1 and DE/randrl/1/exp are used more frequently than DE/rand/1/bin. The typical development is shown for f11 in Fig. 1e.
- f9 – At the beginning, the overall success rate is about 10 % and increases up to 50 % in the middle, then decreases to zero because the correct solution is found, see Fig. 1f. DE/randrl/1/exp is the most frequent strategy.
- f8, f14 – Very low success rate at the beginning and even decreasing over the whole search. The typical development is shown for f14 in Fig. 1g.

7. Conclusion

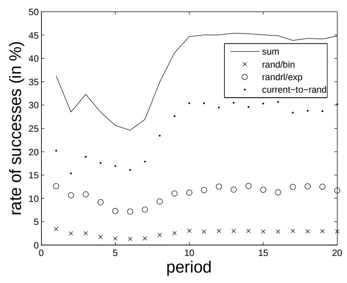
A new adaptive DE algorithm (*comp3jDE*) combining two adaptive mechanisms successfully applied in former experimental studies [2, 17] was proposed and experimentally tested. The novel algorithm outperformed the standard *jDE* algorithm in several CEC2005 benchmark problems but its efficiency on average was worse compared to the standard *jDE* and *b6e6rl* variant of adaptive DE.

The combination of strategies selected to *comp3jDE* was appropriate for solving some problems but not suitable for others. The different development of strategy success for various benchmark functions indicates that the combination of the three strategies chosen to this variant is too greedy for some problems, where the total

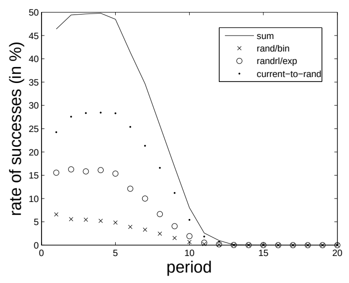
a) function f6



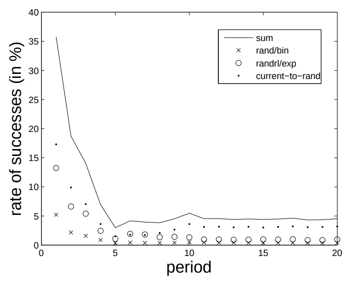
b) function f18



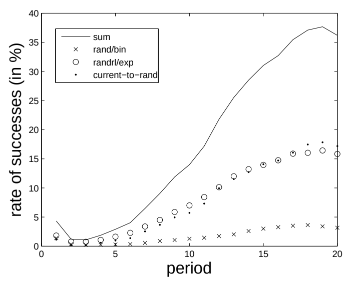
c) function f1



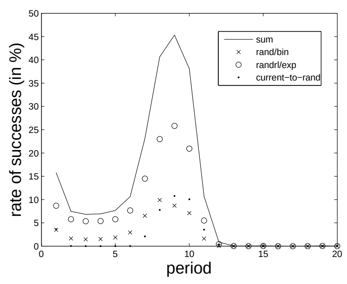
d) function f23



e) function f11



f) function f9



g) function f14

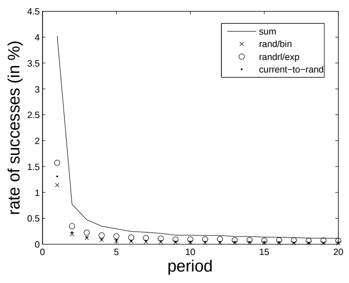


Fig. 1 *The development of rates of DE strategy successes in comp3jDE for selected functions.*

success rates decrease to very small values. Due to the fact the search process tends to the stagnation because the population is not able to move towards better candidates of the solution and continuation of search becomes ineffective. Nevertheless, a combination of competing strategies within the *jDE* frame of evolutionary control-parameter adaptation is a promising way towards new adaptive DE variants. A proper selection of suitable strategies into such combined adaptive model will be addressed in our future research.

Acknowledgement

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 partly supported by University of Ostrava, the project SGS13/PřF/2012.

References

[1] Back T.: Evolutionary algorithms in theory and practice, Oxford University Press, New York, 1996.

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

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

[4] Das S., Abraham A., Chakraborty U. K., Konar A.: Differential evolution using a neighborhood-based mutation operator, IEEE Transactions on Evolutionary Computation, 13, 2009, pp. 526–553.

[5] Hrstka O., Kučerová A.: Improvements of real coded genetic algorithms based on differential operators preventing premature convergence, Advances in Engineering Software, 35, 2004, pp. 237–246.

[6] Kaelo P., Ali M. M.: A numerical study of some modified differential evolution algorithms, European J. Operational Research, 169, 2006, pp. 1176–1184.

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

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

[9] Poláková R., Tvrdík J.: Competitive differential evolution algorithm in comparison with other adaptive variants, 7th International Conference SOCO'12, 2012, pp. 133–142.

[10] Price K. V., Storn R., Lampinen J.: Differential evolution: A practical approach to global optimization, Springer, 2005.

[11] Qin A. K., Huang V. L., Suganthan P. N.: Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Transactions on Evolutionary Computation, 13, 2009, pp. 398–417.

[12] Rahnamayan S., Tishoosh H. R., Salama M. M. A.: Opposition-based differential evolution, IEEE Transactions on Evolutionary Computation, 12, 2008, pp. 64–79.

[13] Storn R., Price K.: Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report, 1995.

[14] Storn R., Price K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Global Optimization, 11, 1997, pp. 341–359.

[15] Suganthan P. N., Hansen N., Liang J. J., Deb K., Chen Y.-P., Auger A., Tiwari S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, Technical Report, 2005, available [Online] http://www.ntu.edu.sg/home/epnsugan/

[16] Tvrdík J.: Competitive differential evolution, MENDEL 2006, 12th International Conference on Soft Computing, 2006, pp. 7–12.

[17] Tvrdík J.: Adaptation in differential evolution: a numerical comparison, Applied Soft Computing, 9, 2009, pp. 1149–1155.

[18] Tvrdík J.: Self-adaptive variants of differential evolution with exponential crossover, Analele of West University Timisoara, Series Mathematics-Informatics, 47, 2009, pp. 151–168, available [Online] http://www1.osu.cz/~tvrdik/down/global_optimization.html

[19] Zaharie D.: A comparative analysis of crossover variants in differential evolution, Proceedings of IMCSIT, 2007, pp. 171–181.

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

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

Appendix

	NP=30		NP=60		NP=90		NP=120	
func.	mean	std	mean	std	mean	std	mean	std
f1	8.94E+00	2.83E+01	3.17E−09	1.58E−08	6.37E−14	1.89E−14	6.59E−14	2.13E−14
f2	6.93E+00	3.06E+01	2.22E−07	8.60E−07	4.37E−09	1.06E−08	1.46E−08	6.48E−08
f3	1.80E+06	8.90E+05	8.95E+05	6.23E+05	5.84E+05	2.71E+05	4.41E+05	2.16E+05
f4	3.72E+03	3.03E+03	9.56E+01	1.81E+02	2.10E−01	5.93E−01	2.89E−03	5.94E−03
f5	5.48E+03	1.16E+03	3.44E+03	6.25E+02	2.67E+03	5.80E+02	2.25E+03	6.08E+02
f6	4.04E+06	1.46E+07	4.45E+04	2.04E+05	2.89E+01	3.48E+01	9.29E+01	3.29E+02
f7	1.64E+00	8.10E+00	2.37E−02	2.55E−02	5.32E−02	1.77E−01	2.54E−02	2.17E−02
f8	2.09E+01	4.66E−02	2.09E+01	4.93E−02	2.09E+01	4.88E−02	2.09E+01	5.96E−02
f9	6.92E+00	9.93E+00	4.38E−01	7.64E−01	4.55E−14	2.32E−14	3.98E−02	1.99E−01
f10	1.02E+02	4.63E+01	5.29E+01	1.20E+01	4.85E+01	1.73E+01	3.60E+01	1.25E+01
f11	2.14E+01	3.02E+00	1.73E+01	2.56E+00	1.41E+01	2.83E+00	1.24E+01	3.99E+00
f12	1.30E+04	8.43E+03	9.36E+03	1.18E+04	5.86E+03	5.44E+03	7.18E+03	9.90E+03
f13	1.27E+00	2.18E−01	1.61E+00	1.37E−01	1.85E+00	1.69E−01	2.05E+00	2.56E−01
f14	1.18E+01	7.34E−01	1.22E+01	5.30E−01	1.26E+01	2.30E−01	1.26E+01	3.47E−01
f15	4.25E+02	7.95E+01	4.05E+02	8.14E+01	4.17E+02	6.84E+01	4.09E+02	6.44E+01
f16	1.69E+02	9.80E+01	1.29E+02	1.21E+02	6.03E+01	1.24E+01	6.70E+01	7.03E+01
f17	2.17E+02	1.69E+02	1.06E+02	7.94E+01	8.82E+01	7.89E+01	7.38E+01	7.67E+01
f18	9.59E+02	4.96E+01	9.16E+02	3.86E+01	8.94E+02	4.82E+01	8.84E+02	4.84E+01
f19	9.48E+02	4.92E+01	9.06E+02	4.85E+01	8.96E+02	4.35E+01	9.02E+02	3.11E+01
f20	9.46E+02	3.67E+01	9.05E+02	4.91E+01	9.16E+02	1.22E+01	8.88E+02	4.53E+01
f21	9.79E+02	3.36E+02	5.93E+02	2.27E+02	5.27E+02	1.34E+02	5.14E+02	7.03E+01
f22	9.45E+02	3.95E+01	9.24E+02	2.17E+01	8.96E+02	2.15E+01	8.99E+02	2.41E+01
f23	9.92E+02	2.43E+02	6.44E+02	1.88E+02	6.12E+02	1.63E+02	5.41E+02	1.23E+01
f24	3.29E+02	3.09E+02	2.00E+02	4.65E−12	2.00E+02	2.23E−12	2.00E+02	1.63E−12
f25	3.17E+02	2.76E+02	2.14E+02	1.68E+00	2.13E+02	1.47E+00	2.12E+02	1.31E+00

Tab. V *Means and standard deviations of the function error values of comp3jDE at different levels of the population size.*

	NP=30		NP=60		NP=90		NP=120	
func.	mean	std	mean	std	mean	std	mean	std
f1	2.96E−14	2.90E−14	2.27E−15	1.14E−14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f2	1.25E−11	1.49E−11	2.22E−08	4.27E−08	2.07E−06	2.08E−06	7.95E−05	1.04E−04
f3	1.55E+05	8.85E+04	1.67E+05	1.15E+05	1.61E+05	7.90E+04	2.10E+05	8.72E+04
f4	2.01E+02	7.45E+02	8.86E−02	1.84E−01	1.72E−01	4.71E−01	1.55E−01	1.72E−01
f5	1.57E+03	5.84E+02	9.29E+02	4.14E+02	5.54E+02	4.85E+02	3.88E+02	2.46E+02
f6	1.12E+00	1.83E+00	1.42E+00	1.85E+00	1.58E+01	2.04E+01	3.25E+01	2.77E+01
f7	2.08E−02	1.89E−02	1.97E−02	1.30E−02	1.28E−02	7.37E−03	8.77E−03	4.27E−03
f8	2.09E+01	6.45E−02	2.09E+01	4.17E−02	2.09E+01	6.14E−02	2.09E+01	5.09E−02
f9	2.79E−01	5.39E−01	3.98E−02	1.99E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f10	3.41E+01	5.88E+00	4.12E+01	6.34E+00	4.89E+01	7.95E+00	5.78E+01	1.00E+01
f11	2.47E+01	3.72E+00	2.64E+01	2.12E+00	2.74E+01	1.61E+00	2.80E+01	1.69E+00
f12	6.60E+03	6.33E+03	6.12E+03	5.90E+03	1.36E+04	7.43E+03	1.35E+04	6.17E+03
f13	9.95E−01	1.13E−01	1.28E+00	1.05E−01	1.45E+00	1.44E−01	1.65E+00	1.21E−01
f14	1.27E+01	3.42E−01	1.28E+01	2.78E−01	1.29E+01	2.06E−01	1.30E+01	1.97E−01
f15	3.40E+02	1.14E+02	3.49E+02	1.02E+02	3.88E+02	5.26E+01	3.60E+02	8.55E+01
f16	1.52E+02	1.45E+02	8.06E+01	6.87E+01	6.98E+01	7.73E+00	8.03E+01	2.85E+01
f17	1.41E+02	1.23E+02	1.10E+02	3.38E+01	1.32E+02	4.20E+01	1.34E+02	1.57E+01
f18	9.08E+02	3.07E+00	9.05E+02	1.26E+00	9.04E+02	8.47E−01	9.05E+02	1.06E+00
f19	9.07E+02	2.45E+00	9.05E+02	1.10E+00	9.05E+02	1.24E+00	9.04E+02	6.13E−01
f20	9.07E+02	2.72E+00	9.05E+02	1.01E+00	9.05E+02	1.05E+00	9.04E+02	8.80E−01
f21	5.00E+02	1.16E−13	5.00E+02	1.16E−13	5.00E+02	1.16E−13	5.00E+02	1.16E−13
f22	8.78E+02	1.69E+01	8.81E+02	1.34E+01	8.76E+02	1.68E+01	8.69E+02	1.95E+01
f23	5.41E+02	3.42E+01	5.34E+02	3.67E−04	5.34E+02	2.53E−04	5.34E+02	2.81E−04
f24	2.00E+02	1.13E−12	2.00E+02	0.00E+00	2.00E+02	0.00E+00	2.00E+02	0.00E+00
f25	2.13E+02	1.75E+00	2.12E+02	1.13E+00	2.11E+02	5.64E−01	2.10E+02	7.49E−01

Tab. VI *Means and standard deviations of the function error values of jDE at different levels of the population size.*