2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

Evaluating the performance of SHADE on CEC 2013 benchmark problems

10.1109/CEC.2013.6557798

Ryoji Tanabe and Alex Fukunaga
Graduate School of Arts and Sciences
The University of Tokyo

***Abstract*—This paper evaluates the performance of Success-History based Adaptive DE (SHADE) on the benchmark set for the CEC2013 Competition on Real-Parameter Single Objective Optimization. SHADE is an adaptive differential algorithm which uses a history-based parameter adaptation scheme. Experimental results on 28 problems from the CEC2013 benchmarks for 10, 30, and 50 dimensions are presented, including measurements of algorithmic complexity. In addition, we investigate the parameter adaptation behavior of SHADE on these instances.**

I. Introduction

The single-objective, global optimization problem is the problem of finding a vector $\boldsymbol{x} = (x_1, ..., x_D)$ which minimizes an objective function $f(\boldsymbol{x})$, where D is the dimensionality of the problem. In *black-box optimization* the task is solving a global optimization problem without explicit knowledge of the form or structure of the objective function, i.e., f is a "black box". This problem occurs in many real-world problems (e.g. engineering optimization where the objective function is computed using a complex simulation), and has been studied by many researchers.

Differential Evolution (DE) is a stochastic, black-box search method that was originally designed for numerical optimization problems [1]. Despite of its relative simplicity, DE has been shown to be competitive with other more complex optimization algorithms, and has been applied to many practical problems [2].

The search efficiency of DE depends largely on the mutation strategy, as well as the control parameters population size, scaling factor F, crossover rate CR, and many previous researchers have shown that optimal settings for these parameters are domain-dependent (c.f. [1], [3], [4], [5]). Thus, there has been much recent work which seeks to automate the selection of the mutation strategy, as well as the control parameter values [6], [2].

Success-History based Adaptive DE (SHADE), which was introduced in a companion paper [7], is a novel, adaptive DE algorithms and an improved version of JADE [8] which uses a different parameter adaptation mechanism based on a history. In SHADE, the mean values of successful parameter CR, F for each generation are stored in a historical memory M_{CR}, M_F. SHADE maintains a diverse set of parameters to guide control parameter adaptation as search progresses for more robust search.

In our companion paper [7], SHADE was evaluated by comparisons with CoDE [9], EPSDE [10], JADE [8] and dynNP-jDE [11] on 28 problems from the CEC2013 benchmark set [12], as well as CEC2005 benchmarks [13] and the set of classical 13 benchmark problems [14]. The experimental results show that SHADE is competitive with previous, state of the art DE algorithms on these benchmarks.

This paper presents an in-depth analysis of the performance evaluation of SHADE on the CEC2013 benchmark set. This benchmark set is improved version of the CEC2005 benchmark set [13]. Experiments were performed for $D = 10$, 30, and 50 dimensional problems.

The rest of the paper is organized as follows. In Section II, we describe the SHADE algorithm. Section III presents our empirical evaluation of SHADE on the CEC2013 benchmarks, which includes empirical measurements of the algorithm complexity (Section III-B), performance results on the benchmark problems (Section III-C), and an analysis of the parameter adaptation behavior of SHADE (Section III-D). Section IV concludes the paper with a discussion and directions for future work.

II. Success-History based Adaptive DE

Success-History based Adaptive DE (SHADE) [7] is an adaptive DE, which uses a historical memory in order to adapt the control parameters F, CR. The overall algorithm is shown in Algorithm 1. Below, we give an overview of the algorithm:

A DE population is represented as a set of real parameter vectors $\boldsymbol{x}_i = (x_1, ..., x_D), i = 1, ..., N$, where N is the population size. At the beginning of the search, the individual vectors in population are initialized randomly.

SHADE maintains a historical memory with H entries for both of the DE control parameters CR and F, M_{CR}, M_F. In the beginning, the contents of $M_{CR,i}, M_{F,i} (i = 1, ..., H)$ are all initialized to 0.5.

In each generation, the DE control parameters F and CR are generated based on the history based parameter adaptation, trial vectors are generated, selection applied, and the historical memory (M_{CR}, M_F) is updated. This process is repeated until some termination criterion is achieved.

In each generation, the control parameters CR_i and F_i used by each individual $\boldsymbol{x}_i$ are generated by first selecting an index r_i randomly from $[1, H]$, and then applying the equations below:

$$CR_i = \text{randn}_i(M_{CR,r_i}, 0.1) \quad (1)$$

$$F_i = \text{randc}_i(M_{F,r_i}, 0.1) \quad (2)$$

Here, $\text{randn}_i(\mu, \sigma^2)$, $\text{randc}_i(\mu, \sigma^2)$ are values selected randomly from normal and Cauchy distributions with mean μ,

Algorithm 1: SHADE

```
   //  Initialization phase
1  G = 0;
2  Initialize population P_0 = (x_{1,0}, ..., x_{N,0}) randomly;
3  Set all values in M_CR, M_F to 0.5;
4  Archive A = ∅;
5  Index counter k = 1;
   //  Main loop
6  while The termination criteria are not met do
7      S_CR = ∅, S_F = ∅;
8      for i = 1 to N do
9          r_i = Select from [1, H] randomly;
10         CR_{i,G} = randn_i(M_{CR,r_i}, 0.1);
11         F_{i,G} = randc_i(M_{F,r_i}, 0.1);
12         p_{i,G} = rand[p_min, 0.2];
13         Generate trial vector u_{i,G} by current-to-pbest/1/bin;
14     end
15     for i = 1 to N do
16         if f(u_{i,G}) ≤ f(x_{i,G}) then
17             x_{i,G+1} = u_{i,G};
18         else
19             x_{i,G+1} = x_{i,G};
20         end
21         if f(u_{i,G}) < f(x_{i,G}) then
22             x_{i,G} → A;
23             CR_{i,G} → S_CR, F_{i,G} → S_F;
24         end
25     end
26     Whenever the size of the archive exceeds |A|, randomly
       selected individuals are deleted so that |A| ≤ |P|;
27     if S_CR ≠ ∅ and S_F ≠ ∅ then
28         Update M_{CR,k}, M_{F,k} based on S_CR, S_F;
29         k++;
30         If k > H, k is set to 1;
31     end
32 end
```

and variance σ^2. In case a value for CR_i outside of $[0, 1]$ is generated, it is replaced by the limit value (0 or 1) closest to the generated value. When $F_i > 1$, F_i is truncated to 1, and when $F_i \leq 0$, Eq. (2) is repeatedly applied to try to generate a valid value.

In each generation, a mutant vector $\boldsymbol{v}_{i,G}$ is generated from an existing population member $\boldsymbol{x}_{i,G}$ by applying some mutation strategy. SHADE uses current-to-*p*best/1 mutation strategy and external archive the same as JADE [8].

$$\begin{aligned}\boldsymbol{v}_{i,G} = \boldsymbol{x}_{i,G} &+ F_i \cdot (\boldsymbol{x}_{pbest,G} - \boldsymbol{x}_{i,G}) \\ &+ F_i \cdot (\boldsymbol{x}_{r1,G} - \tilde{\boldsymbol{x}}_{r2,G})\end{aligned} \tag{3}$$

In Eq. (3), the individual $\boldsymbol{x}_{pbest,G}$ is randomly selected from the top $N \times p$ ($p \in [0, 1]$) members in the G-th generation. F_i is the F parameter used by individual $\boldsymbol{x}_i$. The greediness of current-to-*p*best/1 depends on the control parameter p in order to balance exploitation and exploration (small p behaves more greedily).

In SHADE, each individual $\boldsymbol{x}_i$ has an associated p_i, which is set according to the equation by generation:

$$p_i = \mathrm{rand}[p_{min}, 0.2] \tag{4}$$

where p_{min} is set such that when the $p$best individual is selected, at least 2 individuals are selected, i.e., $p_{min} = 2/N$. The maximum value 0.2 in Eq. (4) is the maximum value of the range for p suggested by Zhang and Sanderson [8].

In Eq. (3), the index r_1 is randomly selected from $[1, N]$ such that they differ from each other as well as i. $\tilde{\boldsymbol{x}}_{r2,G}$ ($\tilde{\boldsymbol{x}}_{r2,G} \neq \boldsymbol{x}_{r1,G} \neq \boldsymbol{x}_{i,G}$) in Eq. (3) is randomly selected from the union of the population $\boldsymbol{P}$ and the external archive $\boldsymbol{A}$, $\boldsymbol{P} \cup \boldsymbol{A}$. Parent vectors $\boldsymbol{x}_{i,G}$ which were worse than the trial vectors $\boldsymbol{u}_{i,G}$ (and are therefore not selected for survival in the standard DE, Eq. 7) are preserved in archive. The size of the archive is set to the same as that of the population, i.e., $|\boldsymbol{A}| = |\boldsymbol{P}|$. Whenever the size of the archive exceeds $|\boldsymbol{A}|$, randomly selected elements are deleted to make space for the newly inserted elements.

For each dimension j, if the mutant vector element $v_{j,i,G}$ is outside the boundaries $[x_j^{min}, x_j^{max}]$, we applied the same correction performed in [8]:

$$v_{j,i,G} = \begin{cases} (x_j^{min} + x_{j,i,G})/2 & \text{if } v_{j,i,G} < x_j^{min} \\ (x_j^{max} + x_{j,i,G})/2 & \text{if } v_{j,i,G} > x_j^{max} \end{cases} \tag{5}$$

After generating the mutant vector $\boldsymbol{v}_{i,G}$, it is crossed with the parent $\boldsymbol{x}_{i,G}$ in order to generate trial vector $\boldsymbol{u}_{i,G}$. Binomial Crossover is used in SHADE and implemented as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } \mathrm{rand}[0,1) \leq CR_i \text{ or } j = j_{rand} \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{6}$$

$\mathrm{rand}[0, 1)$ denotes a uniformly selected random number from $[0, 1)$, and j_{rand} is a decision variable index which is uniformly randomly selected from $[1, D]$. CR_i is the CR parameter used by individual $\boldsymbol{x}_i$.

After all of the trial vectors $\boldsymbol{u}_{i,G}, 0 \leq i \leq N$ have been generated, a selection process determines the survivors for the next generation. The selection operator in standard DE compares each individual $\boldsymbol{x}_{i,G}$ against its corresponding trial vector $\boldsymbol{u}_{i,G}$, keeping the better vector in the population.

$$\boldsymbol{x}_{i,G+1} = \begin{cases} \boldsymbol{u}_{i,G} & \text{if } f(\boldsymbol{u}_{i,G}) \leq f(\boldsymbol{x}_{i,G}) \\ \boldsymbol{x}_{i,G} & \text{otherwise} \end{cases} \tag{7}$$

As with JADE, the CR_i and F_i values used by successful individuals in Eq. (7) are recorded in S_{CR} and S_F, and at the end of the generation, the contents of memory are updated as follows:

$$M_{CR,k,G+1} = \begin{cases} \mathrm{mean}_{WA}(S_{CR}) & \text{if } S_{CR} \neq \emptyset \\ M_{CR,k,G} & \text{otherwise} \end{cases} \tag{8}$$

$$M_{F,k,G+1} = \begin{cases} \mathrm{mean}_{WL}(S_F) & \text{if } S_F \neq \emptyset \\ M_{F,k,G} & \text{otherwise} \end{cases} \tag{9}$$

An index k ($1 \leq k \leq H$) determines the position in the memory to update. At the beginning of the search k is initialized to 1. k is incremented whenever a new element is inserted into the history. If $k > H$, k is set to 1. In generation G, the k-th element in the memory is updated. In the update equations (8) and (9), note that when all individuals in generation G fail to generate a trial vector which is better than the parent, i.e., $S_{CR} = S_F = \emptyset$, the memory is not updated.

In the update equation (8), the weighted arithmetic mean $\mathrm{mean}_{WA}(S_{CR})$ is computed according to Equation (10) by

Peng et al [15].

$$\text{mean}_{WA}(S_{CR}) = \sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR,k} \quad (10)$$

$$w_k = \frac{\Delta f_k}{\sum_{k=1}^{|S_{CR}|} \Delta f_k} \quad (11)$$

where $\Delta f_k = |f(\boldsymbol{u}_{k,G}) - f(\boldsymbol{x}_{k,G})|$ in Eq. (11).

The weighted Lehmer mean $\text{mean}_{WL}(S_F)$ in the update equation (9) is computed using the formula below, and as with $\text{mean}_{WA}(S_{CR})$, the amount of improvement is used in order to influence the parameter adaptation.

$$\text{mean}_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}} \quad (12)$$

III. Experimental Evaluation of SHADE on CEC2013 Competition Problems

We evaluated the performance of SHADE on the *CEC2013 Special Session on Real-Parameter Single Objective Optimization* benchmark suite [12]. This benchmark set is consist of 28 scalable test functions where the global optimum is shifted to o, $o = (o_1, ...o_D)$ (D is the number of dimensions).

These problems are based on an earlier set of benchmarks, the CEC2005 benchmark set [13]. In addition to introducing some entirely new functions, a major difference from the CEC2005 benchmark set is that for the composite functions that are derived by combining multiple functions, the base functions that are combined are selected more carefully, and an effort was made to make the regions around local optima be continuous.

Functions $F_1 \sim F_5$ are unimodal. $F_6 \sim F_{20}$ are multimodal. F_{17} and F_{18} are double-funnel functions which have high deceptive structure [16]. $F_{21} \sim F_{28}$ are composite functions which combine multiple test problems into a complex landscape. See [12] for details.

A. Algorithm Parameters

This section describes the parameter setting of SHADE and its execution environment. The control parameters we set for SHADE are:

- Population size $N = 100$, and
- memory size $H = N = 100$

These values were used for all experiments in this paper (same for $D = 10, 30, 50$).

The value $N = 100$ was chosen because it was used by several other authors in recent work [17], [8], so we have not tuned N. The memory size $H = 100$ was chosen because it setting the historical memory size to be the equivalent to the population size seemed like a natural choice. An evaluation of $H \in \{5, 10, 30, 50, 100, 200, 300, 400, 500\}$ confirmed that $H = 100$ resulted in good overall performance [7].

Although the standard DE parameters F (scaling factor) and CR (crossover rate) are used internally by SHADE, these parameters are automatically tuned by the history-based adaptation mechanism and are not set by the user.

TABLE I: Algorithm Complexity

	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
$D = 10$	0.0009	0.5144	0.6089	99.7084
$D = 30$		1.5479	1.7198	181.4152
$D = 50$		2.6709	2.8802	220.7530

B. Algorithm Complexity

This section reports "algorithm complexity" of our SHADE code as defined in [12].

All experiments were executed on the following system:

- OS: Ubuntu 12.04 LTS
- CPU: core i7 (2.20GHz)
- RAM: 8GB
- Language: C++
- Compiler: g++ (gcc) with -O3 optimization flag

The algorithm complexity of SHADE was calculated according to the detailed instructions in [12]. Table I shows the computed algorithm complexity on 10, 30 and 50 dimensions. As defined in [12], $T0$ is the time calculated by running the following test problem:

```
for i=1:1000000
 x=0.55+(double)i; x=x+x; x=x./2; x=x*x;
 x=sqrt(x); x=log(x); x=exp(x); y=x/x;
end
```

$T1$ is the time to execute 200,000 evaluations of benchmark function f_{14} by itself with D dimensions, and $T2$ is the time to execute SHADE with 200,000 evaluations of f_{14} in D dimensions. $\hat{T}2$ is the mean $T2$ values of 5 runs. According to Table I, both $T1$ and $\hat{T}2$ scaled linearly with the number of dimensions, as shown by the linear growth of $(T2 - t1)/T0$.

C. Results

We followed the rules of the CEC2013 benchmark competition [12]. For all of the problems, the size of the search space is $[-100, 100]^D$. When the gap between the error values of the best solution found and the optimal solution was 10^{-8} or smaller, the error was treated as 0. The three different dimensions $D = 10, 30, 50$ are tested. The maximum number of objective function calls per run was $D \times 10,000$. The number of runs per problem was 51, and we recorded best, worst, median, mean and standard deviation of the error value between the best fitness values found in each run and the true optimal value. The results on $D = 10, 30$ and 50 dimensions are shown in Table II, III and IV respectively.

We have also plotted average error (compared to the optimal solution) vs. the number of evaluations (for $D = 10, 30, 50$ dimensions) in Figures 1, 2 and 3. Note that the y-axis (average error value) is logarithmic scale.

For functions F_1, F_5 and F_{11}, SHADE found the optimal solution in every single run for $D = 10$, $D = 30$, and $D =$

50. This is most likely because these functions are separable, which results in a easy search space for DE.

In several problems, including F_{17}, F_{26}, F_{28} ($D = 10, 30, 50$), and F_6 ($D = 50$), we noticed that SHADE converged to the same local optimum in most every single run. This may be because these functions have a particularly deceptive landscape structure.

As shown in Tables II, III and IV and Figures 1, 2 and 3, the search performance of SHADE usually degrades as the number of dimensions (D) increases.

However, there are interesting exceptions. For F_6 and F_{21}, the results on 30 dimensions is better than the results for 10 dimensions on these functions. Similarly, on F_{22}, SHADE performed better in for $D = 50$ dimensions than for $D = 30$ dimensions. It has been noted by Whitley et al that for some scalable benchmark functions, finding the global optimum with a higher number of dimensions is easier than finding the global optimum on a smaller number of dimensions [18]. It is possible that our anomalous results with F_6, F_{21}, and F_{22} are related to these previous observations.

D. Behavior of Adaptive Parameters

This section investigates the behavior of the DE control parameters that are adapted by SHADE. Figure 4 shows the how the F (scaling factor) and CR (crossover rate) parameters that are automatically tuned by SHADE change over time on benchmark problems $F_1, F_4, F_{11}, F_{14}, F_{24}, F_{27} (D = 30)$. The plots start at the beginning of the search, and continue until SHADE converges (search progress stops).

Each of these figures represents 1 run of SHADE on a benchmark problem. In each figure, the F and CR values for all elements in the historical memory are shown. In all cases, we observe that the values in memory are clustered in a band that moves coherently as the search progresses.

The F and CR parameters converge to different sets of values, depending on the benchmark problem. This is consistent with observation by previous researchers that optimal settings for these control parameters are domain-dependent (c.f. [1], [3], [4], [5]).

One notable pattern can be observed. For the separable and multimodal functions (F_{11} and F_{14}), the control parameters adapt so that F is high and CR converges to relatively low values, while on the nonseparable functions F_4, F_{24}, F_{27}, the control parameters adapt so that CR is very high and F has a moderate value.

IV. Conclusions

This paper reported the performance evaluation of Success-History based Adaptive DE (SHADE) [7] on the CEC2013 benchmark set. Results for $D = 10, 30, 50$, including algorithm complexity measurements were reported. We also analyzed the parameter adaptation behavior of SHADE.

Although we have shown in a companion paper that SHADE significantly outperforms previous DE variants [7], a detailed analysis of SHADE's behavior on the CEC2013 benchmarks shows that for some problems, SHADE repeatedly converges to one particular local optimum. Identifying the cause of this phenomenon and improving SHADE to avoid always being attracted to a particular local optimum is an area of future work.

Finally, we noted that there are some problems which appear to be easier for SHADE when the number of dimensions is smaller. Further understanding of this anomaly is an area for future work.

TABLE II: Results for $D = 10$

Func.	Best	Worst	Median	Mean	Std.
1	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
2	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
3	0.0000e+00	6.3150e+00	0.0000e+00	1.2663e-01	8.8399e-01
4	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
5	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
6	0.0000e+00	9.8124e+00	9.8124e+00	7.8884e+00	3.9346e+00
7	9.5226e-05	2.4897e-02	1.4915e-03	3.2593e-03	4.5447e-03
8	2.0120e+01	2.0488e+01	2.0369e+01	2.0353e+01	8.9488e-02
9	1.1600e+00	4.5580e+00	3.5258e+00	3.3895e+00	7.3507e-01
10	0.0000e+00	3.1293e-02	9.1545e-03	1.1987e-02	8.9885e-03
11	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
12	1.2161e+00	5.2012e+00	3.2817e+00	3.1413e+00	9.7343e-01
13	1.0962e+00	8.6912e+00	3.5469e+00	3.7708e+00	1.8530e+00
14	0.0000e+00	6.2454e-02	0.0000e+00	4.8984e-03	1.6958e-02
15	1.9595e+02	6.5915e+02	4.3993e+02	4.2075e+02	1.1444e+02
16	3.6202e-01	1.2787e+00	6.5561e-01	7.0799e-01	2.1188e-01
17	1.0122e+01	1.0122e+01	1.0122e+01	1.0122e+01	0.0000e+00
18	1.2911e+01	2.0465e+01	1.6775e+01	1.6878e+01	1.5363e+00
19	2.4592e-01	4.4723e-01	3.4914e-01	3.4352e-01	4.8987e-02
20	1.4155e+00	2.9698e+00	2.1117e+00	2.1572e+00	3.5157e-01
21	4.0019e+02	4.0019e+02	4.0019e+02	4.0019e+02	0.0000e+00
22	2.4467e-06	2.0529e+01	8.0390e-01	4.8396e+00	6.1972e+00
23	1.1922e+02	7.8669e+02	4.8381e+02	4.6061e+02	1.7839e+02
24	1.0576e+02	2.0679e+02	2.0000e+02	1.9344e+02	2.4638e+01
25	2.0000e+02	2.0424e+02	2.0000e+02	2.0015e+02	7.0243e-01
26	1.0125e+02	2.0002e+02	1.0619e+02	1.3333e+02	4.3581e+01
27	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	1.4604e-08
28	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	0.0000e+00

TABLE III: Results for $D = 30$

Func.	Best	Worst	Median	Mean	Std.
1	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
2	1.2842e+03	4.1728e+04	6.7037e+03	9.0022e+03	7.4746e+03
3	0.0000e+00	1.5000e+03	1.5249e-04	4.0206e+01	2.1298e+02
4	5.7095e-07	1.2831e-03	7.4687e-05	1.9216e-04	3.0100e-04
5	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
6	0.0000e+00	2.6407e+01	0.0000e+00	5.9596e-01	3.7286e+00
7	4.2101e-01	2.6482e+01	2.5300e+00	4.6016e+00	5.3852e+00
8	2.0439e+01	2.1000e+01	2.0732e+01	2.0723e+01	1.7608e-01
9	2.2707e+01	3.0889e+01	2.7519e+01	2.7466e+01	1.7694e+00
10	1.7236e-02	1.8974e-01	7.1416e-02	7.6855e-02	3.5751e-02
11	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
12	1.5092e+01	3.2652e+01	2.2989e+01	2.3039e+01	3.7320e+00
13	2.3594e+01	8.8801e+01	5.0733e+01	5.0337e+01	1.3371e+01
14	0.0000e+00	1.0410e-01	2.0819e-02	3.1841e-02	2.3315e-02
15	2.5418e+03	3.9055e+03	3.2363e+03	3.2179e+03	2.6367e+02
16	9.7759e-02	1.2039e+00	9.3896e-01	9.1315e-01	1.8549e-01
17	3.0434e+01	3.0434e+01	3.0434e+01	3.0434e+01	3.8300e-14
18	5.9875e+01	8.4548e+01	7.2844e+01	7.2457e+01	5.5837e+00
19	9.6271e-01	1.5258e+00	1.3811e+00	1.3557e+00	1.2013e-01
20	9.3862e+00	1.2128e+01	1.0437e+01	1.0473e+01	6.0435e-01
21	2.0000e+02	4.4354e+02	3.0000e+02	3.0904e+02	5.6473e+01
22	1.2528e+01	1.1344e+02	1.0599e+02	9.8095e+01	2.5213e+01
23	2.4634e+03	4.3517e+03	3.5402e+03	3.5078e+03	4.1093e+02
24	2.0054e+02	2.2239e+02	2.0332e+02	2.0525e+02	5.2904e+00
25	2.0031e+02	2.9217e+02	2.6163e+02	2.5944e+02	1.9645e+01
26	2.0000e+02	3.0601e+02	2.0000e+02	2.0208e+02	1.4844e+01
27	3.0638e+02	8.6714e+02	3.5512e+02	3.8763e+02	1.0897e+02
28	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	0.0000e+00

TABLE IV: Results for $D = 50$

Func.	Best	Worst	Median	Mean	Std.
1	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
2	7.1372e+03	5.3130e+04	2.6821e+04	2.6565e+04	1.1300e+04
3	1.9530e+00	9.9033e+06	2.9846e+05	8.7997e+05	1.9583e+06
4	3.0589e-05	6.8969e-03	1.1310e-03	1.6131e-03	1.4141e-03
5	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
6	4.5539e+00	4.9127e+01	4.3447e+01	4.2796e+01	5.5197e+00
7	8.9970e+00	4.8966e+01	2.2616e+01	2.3296e+01	9.3203e+00
8	2.0599e+01	2.1218e+01	2.0862e+01	2.0905e+01	1.6831e-01
9	5.0202e+01	5.8880e+01	5.5710e+01	5.5424e+01	1.9838e+00
10	7.3960e-03	1.8721e-01	6.6532e-02	7.3658e-02	3.6745e-02
11	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
12	3.7373e+01	8.6039e+01	5.7777e+01	5.8632e+01	1.1128e+01
13	1.1086e+02	1.8693e+02	1.4084e+02	1.4538e+02	1.9515e+01
14	0.0000e+00	7.4950e-02	2.4983e-02	3.4536e-02	1.9286e-02
15	5.7513e+03	7.6092e+03	6.8016e+03	6.8203e+03	4.4061e+02
16	7.3791e-01	1.7119e+00	1.2904e+00	1.2821e+00	2.0717e-01
17	5.0786e+01	5.0786e+01	5.0786e+01	5.0786e+01	4.2668e-14
18	1.0509e+02	1.6692e+02	1.3588e+02	1.3679e+02	1.2906e+01
19	1.8845e+00	3.2937e+00	2.6798e+00	2.6414e+00	2.8329e-01
20	1.7610e+01	2.0681e+01	1.9361e+01	1.9271e+01	7.6995e-01
21	2.0000e+02	1.1224e+03	1.1222e+03	8.4485e+02	3.6263e+02
22	9.0382e+00	5.4272e+01	1.1645e+01	1.3263e+01	7.1167e+00
23	5.8816e+03	9.0177e+03	7.6280e+03	7.6289e+03	6.5792e+02
24	2.1652e+02	2.5768e+02	2.3246e+02	2.3384e+02	1.0145e+01
25	2.7557e+02	3.7893e+02	3.4952e+02	3.3964e+02	3.0862e+01
26	2.0000e+02	4.1906e+02	2.0162e+02	2.5751e+02	8.0778e+01
27	5.2911e+02	1.7083e+03	8.4616e+02	9.3554e+02	3.0665e+02
28	4.0000e+02	3.3500e+03	4.0000e+02	4.5784e+02	4.1309e+02

REFERENCES

[1] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[2] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Tran. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, 2011.

[3] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," in *Int. Conf. on Adv. in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, 2002, pp. 293–298.

[4] J. Rönkkönen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *IEEE CEC*, 2005, pp. 506–513.

[5] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *GECCO*, 2006, pp. 485–492.

[6] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.

[7] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *submission to IEEE CEC, available at http://metahack.org/cec2013de.pdf*, 2013.

[8] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Tran. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, 2009.

[9] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Tran. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, 2011.

[10] 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, no. 2, pp. 1679–1696, 2011.

[11] J. Brest and M. S. Maučec, "Population size reduction for the differential evolution algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.

[12] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," Nanyang Technological University, Tech. Rep., 2013.

[13] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," Nanyang Technological University, Tech. Rep., 2005.

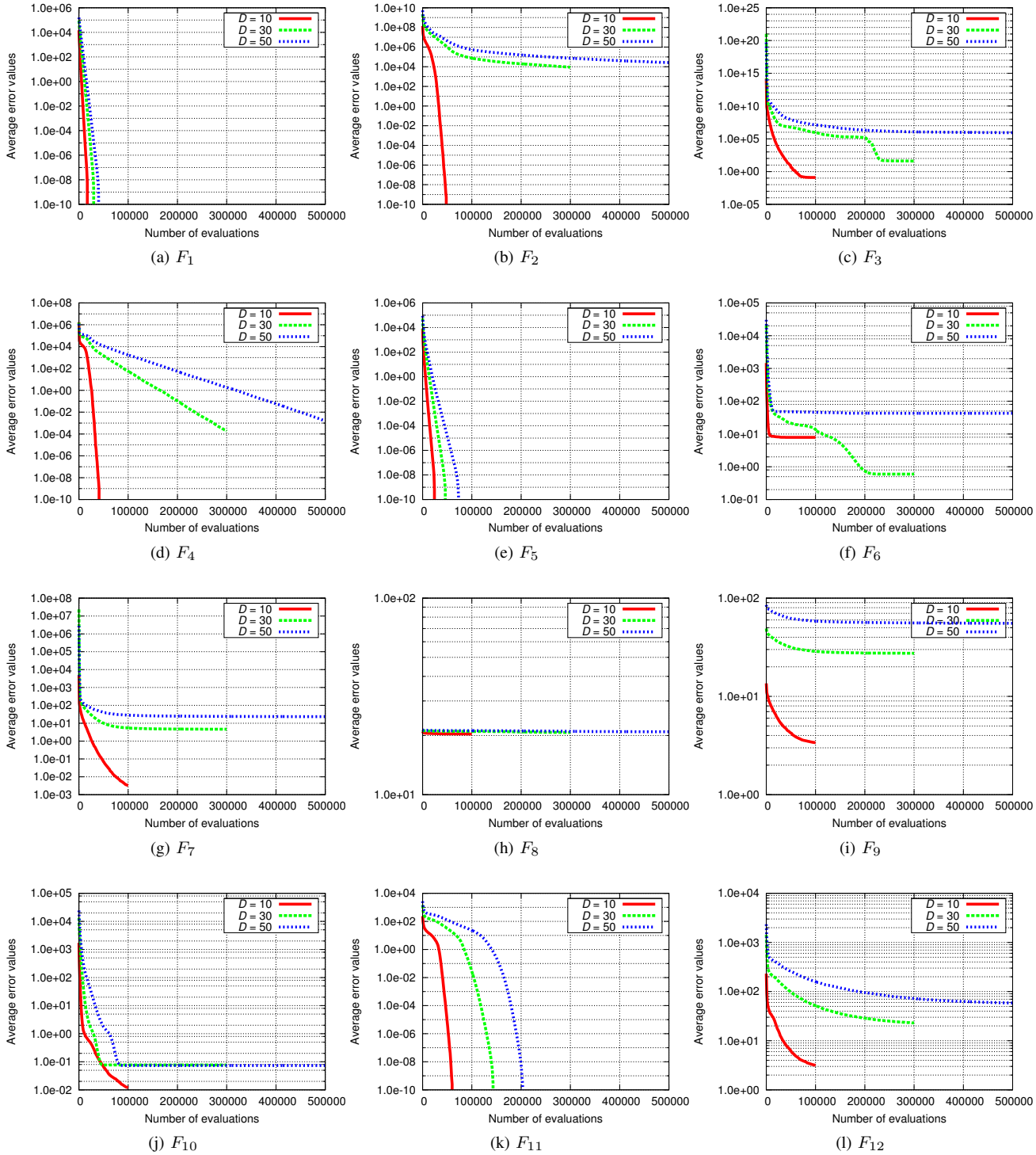
[14] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Tran. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, 1999.

[15] F. Peng, K. Tang, G. Chen, and X. Yao, "Multi-start JADE with knowledge transfer for numerical optimization," in *IEEE CEC*, 2009, pp. 1889–1895.

[16] M. Lunacek, D. Whitley, and A. Sutton, "The impact of global structure on search," in *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 498–507.

[17] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Tran. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, 2006.

[18] D. Whitley, K. Mathias, S. Rana, and J. Dzubera, "Evaluating evolutionary algorithms," *Artificial Intelligence*, vol. 85, pp. 245–276, 1996.



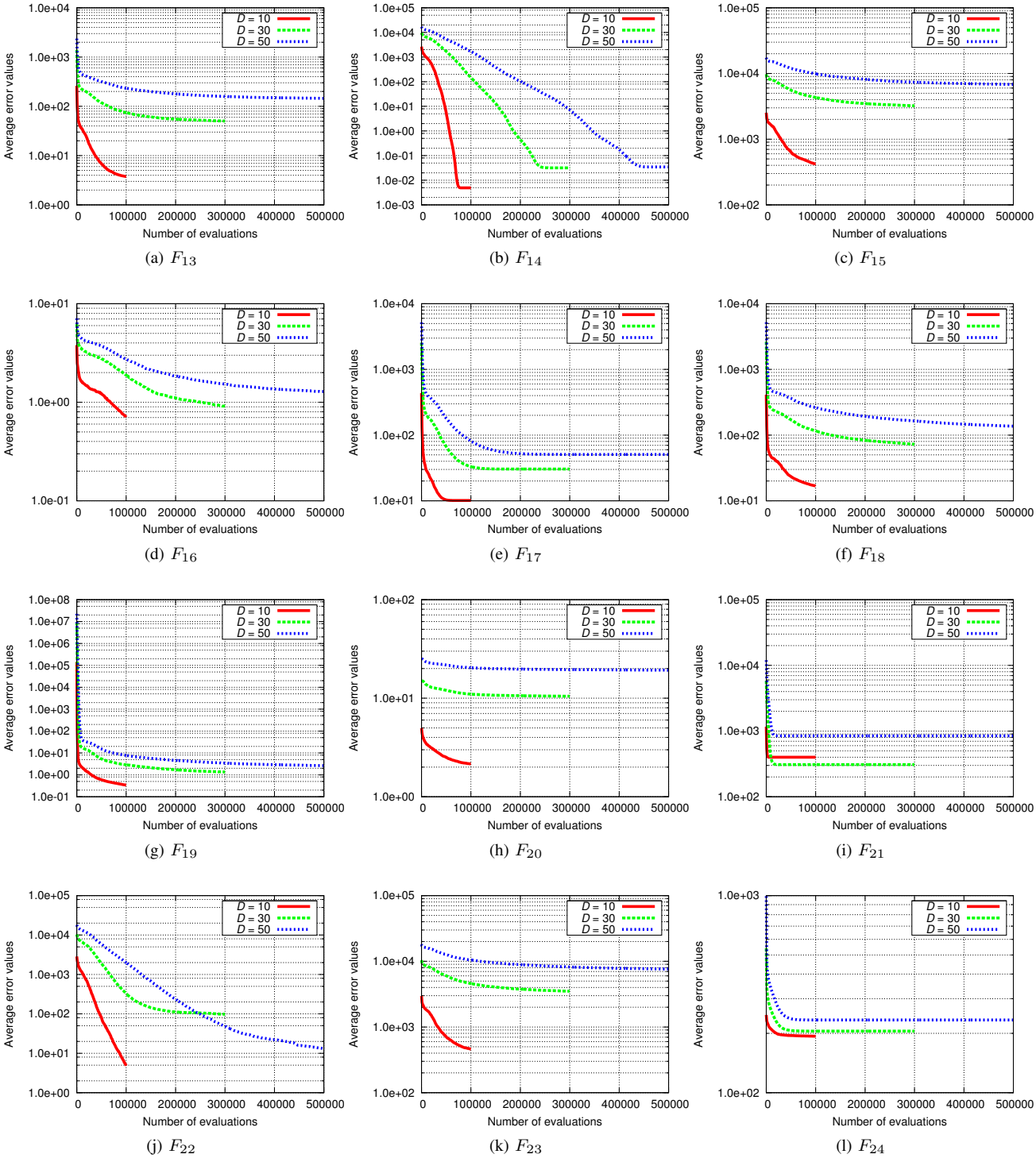
(a) F_1 (b) F_2 (c) F_3

(d) F_4 (e) F_5 (f) F_6

(g) F_7 (h) F_8 (i) F_9

(j) F_{10} (k) F_{11} (l) F_{12}

Fig. 1: Convergence graph $f_1 \sim f_{12}$



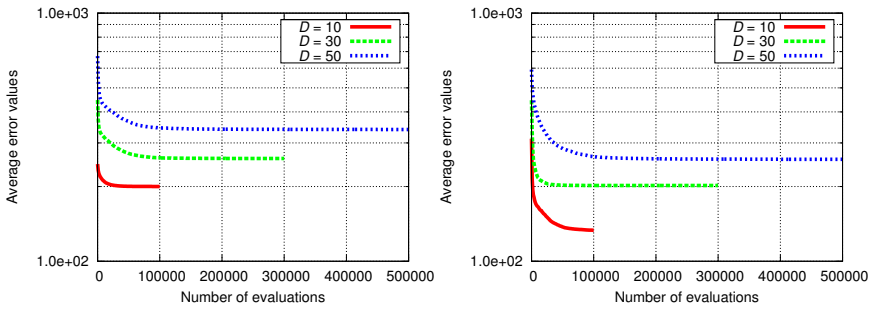
(a) F_{13} (b) F_{14} (c) F_{15}

(d) F_{16} (e) F_{17} (f) F_{18}

(g) F_{19} (h) F_{20} (i) F_{21}

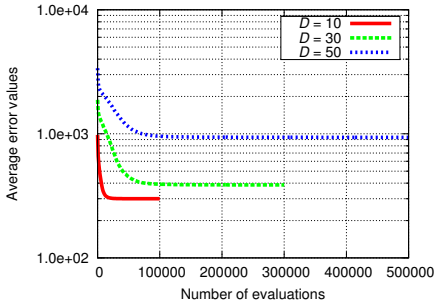
(j) F_{22} (k) F_{23} (l) F_{24}

Fig. 2: Convergence graph $f_{13} \sim f_{24}$

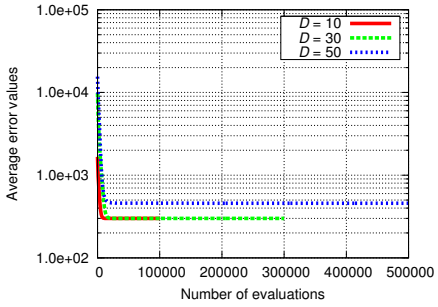


(a) F_{25}

(b) F_{26}

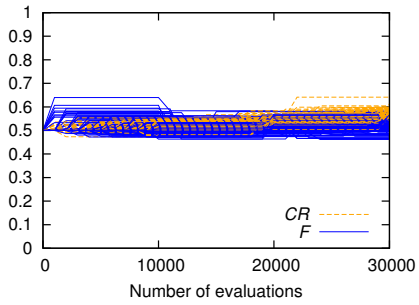


(c) F_{27}

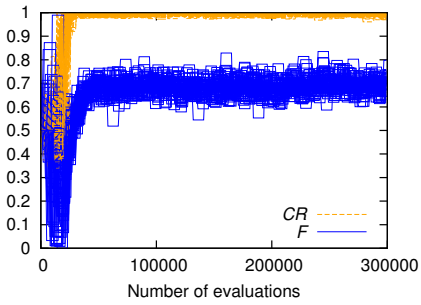


(d) F_{28}

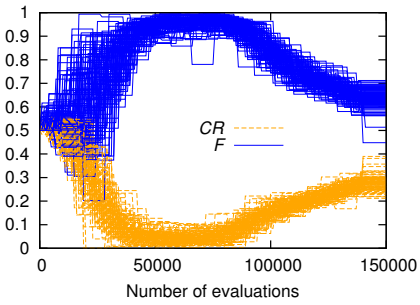
Fig. 3: Convergence graph $f_{25} \sim f_{28}$



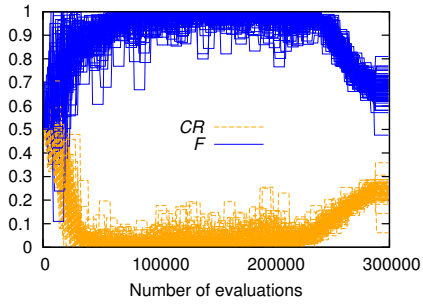
(a) F_1



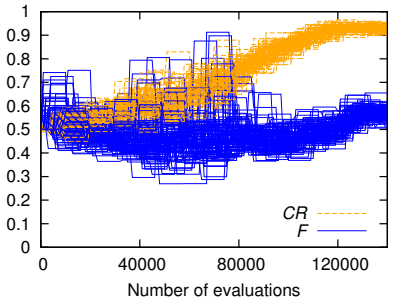
(b) F_4



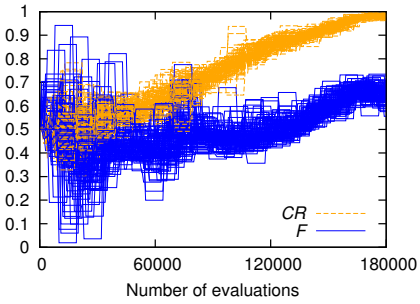
(c) F_{11}



(d) F_{14}



(e) F_{24}



(f) F_{27}

Fig. 4: Parameter adaptation process of SHADE on $F_1, F_4, F_{11}, F_{14}, F_{24}, F_{27}$.