Generalized Lehmer Mean for Success History based Adaptive Differential Evolution

Vladimir Stanovov[1,2][a], Shakhnaz Akhmedova[1][b], Eugene Semenkin[2][c] and Mariia Semenkina[3][d]

[1]*Reshetnev Siberian State University, Krasnoyarskii rabochii ave. 31, 660037, Krasnoyarsk, Russian Federation*

[2]*Siberian Federal University, Institute of Mathematics and Computer Science,*
79 Svobodny pr., 660041, Krasnoyarsk, Russian Federation

[3]*Heuristic and Evolutionary Algorithms Laboratory (HEAL), University of Applied Sciences Upper Austria,*
Softwarepark 11, 4232 Hagenberg, Austria

Keywords: Differential Evolution, Optimization, Parameter Control, Metaheuristic.

Abstract: The Differential Evolution (DE) is a highly competitive numerical optimization algorithm, with a small number of control parameters. However, it is highly sensitive to the setting of these parameters, which inspired many researchers to develop adaptation strategies. One of them is the popular Success-History based Adaptation (SHA) mechanism, which significantly improves the DE performance. In this study, the focus is on the choice of the metaparameters of the SHA, namely the settings of the Lehmer mean coefficients for scaling factor and crossover rate memory cells update. The experiments are performed on the LSHADE algorithm and the Congress on Evolutionary Computation competition on numerical optimization functions set. The results demonstrate that for larger dimensions the SHA mechanism with modified Lehmer mean allows a significant improvement of the algorithm efficiency. The theoretical considerations of the generalized Lehmer mean could be also applied to other adaptive mechanisms.

1 INTRODUCTION

In recent decades the area of Evolutionary Computation (EC) has proposed various tools to solve complex optimization problems in different areas, including engineering, technical, financial, and many more. Most of these tools are based on various biology-inspired heuristic approaches, but mostly follow the evolutionary paradigm. The existing heuristic and metaheuristic algorithms differ in problem solution representation and operations used. One of the well-developed areas is the area of numerical optimization problems, in which certain success has been made by algorithms such as real-coded Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) and Differential Evolution (DE).

The DE is known to be one of the most competitive approaches for all types of optimization problems and since its development (Storn and Price, 1997) has received many attention from the research community due to its simplicity in implementation and only few control parameters. However, the choice of optimal parameter settings remains one of the main issues of modern DE variants development, according to (Das et al., 2016).

In (Al-Dabbagh et al., 2018) it is stated that one of the most efficient parameter adaptation schemes proposed for DE is the Success History based Adaptation, proposed in (Tanabe and Fukunaga, 2013). This adaptation mechanism was applied in one of the most competitive DE variants, such as SHADE, LSHADE, and their modifications. The SHA uses information about previously successful values of DE parameters to update the memory cells. To address the effect of bias caused by the fact that smaller scaling factor and crossover rate parameter values lead to more greedy search, the usage of Lehmer mean was proposed. Some aspects of structural bias in DE have been studied in (Caraffini et al., 2019). In this study, the generalized variant of Lehmer mean is considered, and the LSHADE variant is tested with various types of Lehmer mean implementations.

[a] https://orcid.org/0000-0002-1695-5798
[b] https://orcid.org/0000-0003-2927-1974
[c] https://orcid.org/0000-0002-3776-5707
[d] https://orcid.org/0000-0002-8228-6924

Stanovov, V., Akhmedova, S., Semenkin, E. and Semenkina, M.
Generalized Lehmer Mean for Success History based Adaptive Differential Evolution.
DOI: 10.5220/0008163600930100
In *Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019)*, pages 93-100
ISBN: 978-989-758-384-1

The rest of the paper is organized as follows: in section 2 the DE state of the art is described, with the main focus on the SHA mechanism, section 3 presents the modified Lehmer mean and theoretical insights, section 4 contains the experimental setup and results, as well as discussion, and section 5 concludes the paper.

2 SUCCESS HISTORY BASED DIFFERENTIAL EVOLUTION

The Differential Evolution was originally proposed by K. Price and R. Storn in (Storn and Price, 1997) for real-valued optimization problems solving. DE is a population-based algorithm, in which the population of NP individuals is represented as $x_{i,j}$, where $i = 1,...,NP$, $j = 1,...,D$, where D is the problem dimension. The goal is to minimize the function $f(X)$ with respect to bound constraints $[xmin_j, xmax_j]$.

The DE starts by initializing the population randomly within the boundaries, and proceeds by performing mutation, crossover and selection (replacement) operations. The mutation operator is the key component of DE, which generates new mutant vector by combining the vectors from the population. There are several mutation strategies known, however, in this study the $current-to-pbest$ strategy, introduced in JADE algorithm (Zhang and Sanderson, 2009), is applied:

$$v_j = x_{i,j} + F*(x_{pb,j} - x_{i,j}) + F*(x_{r1,j} - x_{r2,j}) \quad (1)$$

where F is the scaling factor, which is a parameter usually in range $[0,1]$. The pb index is chosen from p% best individuals in the population, while $r1$ and $r2$ are chosen randomly from the population. Next, the crossover operation is performed, in which the trial vector is defined as:

$$u_j = \begin{cases} v_j, & \text{if } rand(0,1) < Cr \text{ or } j = jrand \\ x_{i,j}, & \text{otherwise} \end{cases}$$

where Cr is the crossover rate in range $[0,1]$, $jrand$ is set to random index in $[1,D]$ and used to make sure that at least one variable is taken from the mutant vector. After this, the selection procedure is applied, and the newly generated trial vector u replaces the target vector x_i if it has at least as good fitness:

$$x_{i,j} = \begin{cases} u_j, & \text{if } f(u_j) \le f(x_{i,j}) \\ x_{i,j}, & \text{otherwise} \end{cases}$$

The scaling factor F, crossover rate Cr, as well as population size NP are three main control parameters if DE. The SHADE algorithm (Tanabe and Fukunaga, 2013) improved the adaptation procedure in JADE (Zhang and Sanderson, 2009) by introducing several memory cells containing best known parameter values combinations, which were then used to generate new trial vectors. Initially there were H memory cells, each containing a pair M_F, M_Cr, which are set to 0.5, and the current memory index h was set to 1. For each mutation and crossover the F and Cr values were generated using Cachy distribution and normal distribution with scale parameter and standard deviation of 0.1 respectively:

$$\begin{cases} F = randc(M_{F,h}, 0.1), \\ Cr = randn(M_{Cr,h}, 0.1) \end{cases}$$

If the newly generated F or Cr is outside the $[0,1]$ interval, then it is generated again until it satisfies this condition. During the selection step, if the trial vector was successful, i.e. better then target vector, the values of F and Cr were stored in S_F and S_Cr, as well as the improvement value $\Delta f_j = |f(u) - f(x_i)|$. After the end of the generation, current h-th memory cell is updated using the weighted Lehmer mean:

$$mean_{wL}(S) = \frac{\sum_{j=1}^{|S|} w_j S_j^2}{\sum_{j=1}^{|S|} w_j S_j} \quad (2)$$

where S is either S_F or S_Cr, and the weight value $w_j = \frac{\Delta f_j}{\sum_{k=1}^{|S|} \Delta f_k}$. The index of memory cell h is incremented every generation and set to 1 if $h = H$. The idea of using several memory cells is to provide more robust parameter adaptation, in which the fluctuations of F and Cr would not influence the searh significantly. The SHA mechanism is sensitive not only to the improvement fact, but also to the value of the improvement, i.e. Δf.

3 GENERALIZED LEHMER MEAN FOR SUCCESS HISTORY ADAPTATION

Originally, the usage of Lehmer mean was proposed in JADE (Zhang and Sanderson, 2009) algorithm for the calculation of F values only. In rJADE (Peng et al., 2009), the weighted procedure for F calculation was proposed, where the Δf_j values were used. Finally, in LSHADE (Tanabe and Fukunaga, 2014) the Lehmer mean was used for both F and Cr calculation.

The generalized from of weighted Lehmer mean could be written as follows (Bullen, 2003):

$$mean_{p,w}(X) = \frac{\sum_{j=1}^{|x|} w_j x_j^p}{\sum_{j=1}^{|x|} w_j x_j^{p-1}} \quad (3)$$

From this equation it could be seen that weighted Lehmer means is a group of means, and by changing the p parameter, other means cold be obtained. In particular, $mean_{0,w}(X)$ is the harmonic mean, $mean_{1,w}(X)$ is arithmetic mean, and $mean_{2,w}(X)$ is also referred to as contraharmonic mean. Other means could be considered by changing the p value, for example, Fig. 1 shows the means obtained by changing p in range $[1,4]$ for x values uniformly distributed in $[0,1]$, all $w_j = 1$.

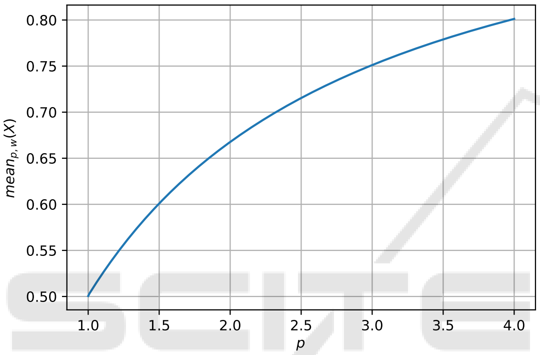


Figure 1: Dependence of generalized Lehmer mean on p parameter.

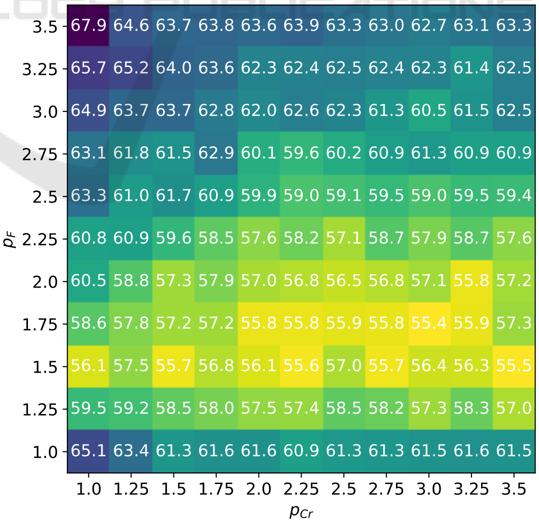
The biased Lehmer mean with large p values could be helpful because it tends to generate larger M_F and M_Cr values, which could be more preferable on a long-term search perspective. This happens because smaller F and Cr result in more greedy search, i.e. for most functions it is easier to generate a solution close to the one at hand, which leads to local search-like process. To determine the optimal p values, the next section contains experimental setup for algorithm performance comparison and results.

4 EXPERIMENTAL SETUP AND RESULTS

To evaluate the performance of LSHADE with different settings of generalized Lehmer mean, a series of experiments has been performed on the set of benchmark problems from the Congress on Evolutionary Computation (CEC) 2017 for single-objective real-valued bound-constrained optimization. The set of benchmark problems consists of 30 functions defined for dimensions 10, 30, 50 and 100. The experiments were performed according to the competition rules, i.e. there were 51 independent run for every function, the total computation resource was set to 10000D, and the best achieved fitness values were saved after NFE = 0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0NFE_{max}, where NFE is current number of function evaluations, and NFE_{max} is the total available resource. The CEC 2017 functions definition can be found in (Wu et al., 2016).

The LSHADE algorithm used the external archive with size NP, and index $r2$ in $current-to-pbest$ mutation strategy was generated from both population and the archive, according to (Tanabe and Fukunaga, 2014). The population size was set to $75D^{\frac{2}{3}}$, p parameter in $current-to-pbest$ strategy equal to 0.17.

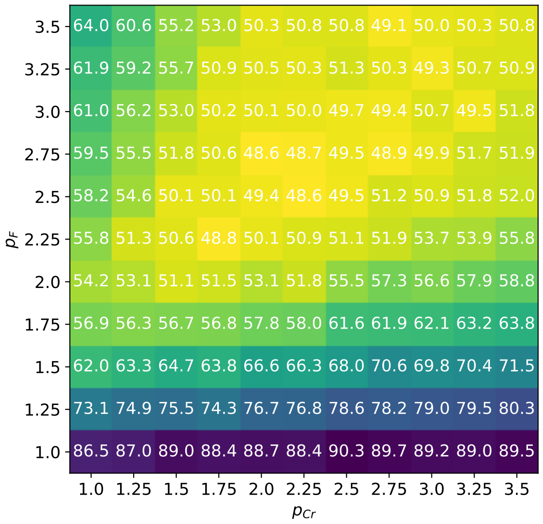
To perform the comprehensive experiment, the LSHADE algorithm was tested with different values of p in Lehmer mean for F and Cr generation. The p_F and p_{Cr} changed between 1 and 3.5 with step 0.25, and the experiments were performed for dimensions 10, 30 and 50. The comparison was performed using two-tailed Mann-Whitney rank sum statistical test with tie-braking and significance level $p = 0.01$ and Friedman ranking procedure. Figure 2, 3 and 4 show the Friedman ranking of different algorithms for 10D, 30D and 50D respectively. The numbers in heatmap graphs represent the truncated to 0.1 precision final ranking.



	1.0	1.25	1.5	1.75	2.0	2.25	2.5	2.75	3.0	3.25	3.5
3.5	67.9	64.6	63.7	63.8	63.6	63.9	63.3	63.0	62.7	63.1	63.3
3.25	65.7	65.2	64.0	63.6	62.3	62.4	62.5	62.4	62.3	61.4	62.5
3.0	64.9	63.7	63.7	62.8	62.0	62.6	62.3	61.3	60.5	61.5	62.5
2.75	63.1	61.8	61.5	62.9	60.1	59.6	60.2	60.9	61.3	60.9	60.9
2.5	63.3	61.0	61.7	60.9	59.9	59.0	59.1	59.5	59.0	59.5	59.4
2.25	60.8	60.9	59.6	58.5	57.6	58.2	57.1	58.7	57.9	58.7	57.6
2.0	60.5	58.8	57.3	57.9	57.0	56.8	56.5	56.8	57.1	55.8	57.2
1.75	58.6	57.8	57.2	57.2	55.8	55.8	55.9	55.8	55.4	55.9	57.3
1.5	56.1	57.5	55.7	56.8	56.1	55.6	57.0	55.7	56.4	56.3	55.5
1.25	59.5	59.2	58.5	58.0	57.5	57.4	58.5	58.2	57.3	58.3	57.0
1.0	65.1	63.4	61.3	61.6	61.6	60.9	61.3	61.3	61.5	61.6	61.5

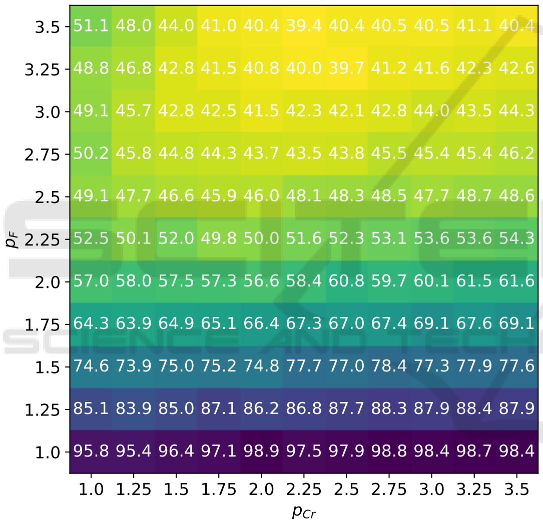
Figure 2: Friedman test, D = 10.

The Friedman ranking procedure performed comparison between all 121 algorithm variants for each dimension. From Figures 2-4 it could be observed that the classical LSHADE setting with $p_F = 2$ and $p_{Cr} = 2$ is not an optimal choice for all dimensions.



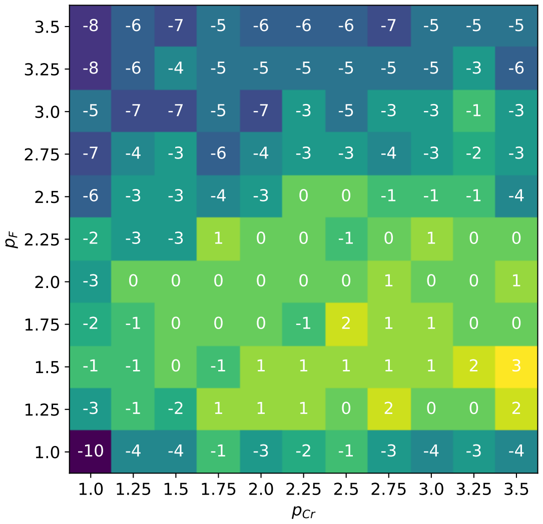
	1.0	1.25	1.5	1.75	2.0	2.25	2.5	2.75	3.0	3.25	3.5
3.5	64.0	60.6	55.2	53.0	50.3	50.8	50.8	49.1	50.0	50.3	50.8
3.25	61.9	59.2	55.7	50.9	50.5	50.3	51.3	50.3	49.3	50.7	50.9
3.0	61.0	56.2	53.0	50.2	50.1	50.0	49.7	49.4	50.7	49.5	51.8
2.75	59.5	55.5	51.8	50.6	48.6	48.7	49.5	48.9	49.9	51.7	51.9
2.5	58.2	54.6	50.1	50.1	49.4	48.6	49.5	51.2	50.9	51.8	52.0
2.25	55.8	51.3	50.6	48.8	50.1	50.9	51.1	51.9	53.7	53.9	55.8
2.0	54.2	53.1	51.1	51.5	53.1	51.8	55.5	57.3	56.6	57.9	58.8
1.75	56.9	56.3	56.7	56.8	57.8	58.0	61.6	61.9	62.1	63.2	63.8
1.5	62.0	63.3	64.7	63.8	66.6	66.3	68.0	70.6	69.8	70.4	71.5
1.25	73.1	74.9	75.5	74.3	76.7	76.8	78.6	78.2	79.0	79.5	80.3
1.0	86.5	87.0	89.0	88.4	88.7	88.4	90.3	89.7	89.2	89.0	89.5

Figure 3: Friedman test, D = 30.



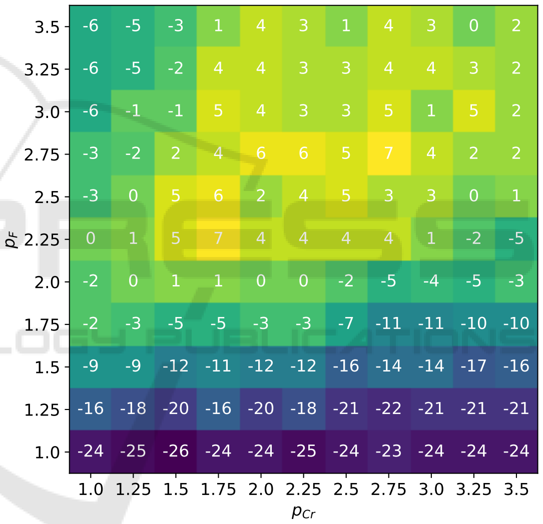
	1.0	1.25	1.5	1.75	2.0	2.25	2.5	2.75	3.0	3.25	3.5
3.5	51.1	48.0	44.0	41.0	40.4	39.4	40.4	40.5	40.5	41.1	40.4
3.25	48.8	46.8	42.8	41.5	40.8	40.0	39.7	41.2	41.6	42.3	42.6
3.0	49.1	45.7	42.8	42.5	41.5	42.3	42.1	42.8	44.0	43.5	44.3
2.75	50.2	45.8	44.8	44.3	43.7	43.5	43.8	45.5	45.4	45.4	46.2
2.5	49.1	47.7	46.6	45.9	46.0	48.1	48.3	48.5	47.7	48.7	48.6
2.25	52.5	50.1	52.0	49.8	50.0	51.6	52.3	53.1	53.6	53.6	54.3
2.0	57.0	58.0	57.5	57.3	56.6	58.4	60.8	59.7	60.1	61.5	61.6
1.75	64.3	63.9	64.9	65.1	66.4	67.3	67.0	67.4	69.1	67.6	69.1
1.5	74.6	73.9	75.0	75.2	74.8	77.7	77.0	78.4	77.3	77.9	77.6
1.25	85.1	83.9	85.0	87.1	86.2	86.8	87.7	88.3	87.9	88.4	87.9
1.0	95.8	95.4	96.4	97.1	98.9	97.5	97.9	98.8	98.4	98.7	98.4

Figure 4: Friedman test, D = 50.



	1.0	1.25	1.5	1.75	2.0	2.25	2.5	2.75	3.0	3.25	3.5
3.5	-8	-6	-7	-5	-6	-6	-6	-7	-5	-5	-5
3.25	-8	-6	-4	-5	-5	-5	-5	-5	-5	-3	-6
3.0	-5	-7	-7	-5	-7	-3	-5	-3	-3	-1	-3
2.75	-7	-4	-3	-6	-4	-3	-3	-4	-3	-2	-3
2.5	-6	-3	-3	-4	-3	0	0	-1	-1	-1	-4
2.25	-2	-3	-3	1	0	0	-1	0	1	0	0
2.0	-3	0	0	0	0	0	0	1	0	0	1
1.75	-2	-1	0	0	0	-1	2	1	1	0	0
1.5	-1	-1	0	-1	1	1	1	1	1	2	3
1.25	-3	-1	-2	1	1	1	0	2	0	0	2
1.0	-10	-4	-4	-1	-3	-2	-1	-3	-4	-3	-4

Figure 5: Mann-Whitney test, D = 10.



	1.0	1.25	1.5	1.75	2.0	2.25	2.5	2.75	3.0	3.25	3.5
3.5	-6	-5	-3	1	4	3	1	4	3	0	2
3.25	-6	-5	-2	4	4	3	3	4	4	3	2
3.0	-6	-1	-1	5	4	3	3	5	1	5	2
2.75	-3	-2	2	4	6	6	5	7	4	2	2
2.5	-3	0	5	6	2	4	5	3	3	0	1
2.25	0	1	5	7	4	4	4	4	1	-2	-5
2.0	-2	0	1	1	0	0	-2	-5	-4	-5	-3
1.75	-2	-3	-5	-5	-3	-3	-7	-11	-11	-10	-10
1.5	-9	-9	-12	-11	-12	-12	-16	-14	-14	-17	-16
1.25	-16	-18	-20	-16	-20	-18	-21	-22	-21	-21	-21
1.0	-24	-25	-26	-24	-24	-25	-24	-23	-24	-24	-24

Figure 6: Mann-Whitney test, D = 30.

Several other important conclusions could be done, for example, for higher dimensions larger p_F are more preferable, probably because exploration properties of the DE are more beneficial in this case. For all dimensions the arithmetical mean, i.e. $p = 1$ is one of the worst possible choices. As for the p_{Cr} values, the dependence on this parameter is not as significant, however, the growth of both p_F and p_{Cr} could deliver good search properties.

Figures 5, 6 and 7 contain the Mann-Whitney test comparison between the case when $p_F = 2$ and $p_{Cr} = 2$ and other cases. The numbers in heatmap graphs represent the total score, which was defined as the sum of wins (+1), ties (0) and losses (-1) of every algorithm compared to baseline variant.

The comparison in Figures 5-7 shows that for $10D$ some of the best variants are around $p_F = 1.5$ and relatively large p_{Cr}. For $30D$ the best choice is $p_F = 2.25$ and $p_{Cr} = 1.75$, where up to 7 significant improvements are found. For $50D$, up to 17 improvements could be achieved with relatively large $p_F = 3.0$ and large $p_{Cr} = 2.0$. From this, it can be concluded that larger dimensional problems require larger F values to be set for a successful search process, while the Cr values are not so significant, although, the values around $p_{Cr} = 2.0$ appear to be a good choice in all cases.

As for $100D$ problems, only a limited set of experiments has been performed due to computational complexity. The Mann-Whitney test comparison between baseline version of LSHADE and LSHADE with variable p_F (LSHADE-p_F) is presented in Ta-

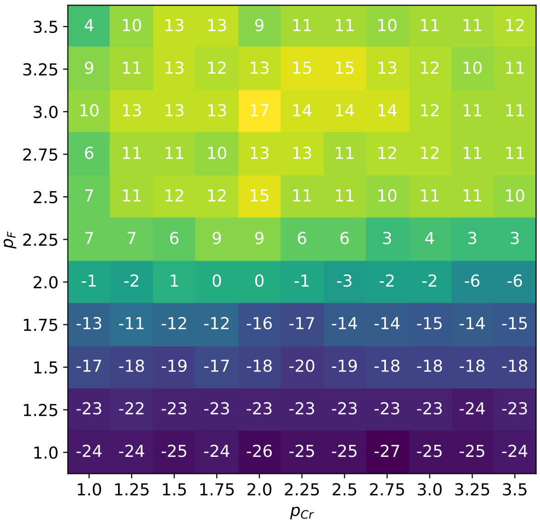


Figure 7: Mann-Whitney test, D = 50.

ble 1. For comparison on 100D the p_F was set to 3.75, and $p_{Cr} = 2.0$, for 10D, 30D and 50D the values of p_F were set to 1.5, 2.25 and 3.0 respectively, with the same p_{Cr} value. Here 1 means that LSHADE-p_F was better for particular function, -1 means that it was worse, and 0 means that there were no significant difference.

The results in Table 1 show that the effect on the DE performance is observed mostly for 50D and 100D. As for different functions, there were improvements for f5 and f8, as well as for f11 and f12, also for f1, f2, f6, f15, f18, f19, f23, f25, f26 and f27. These functions have different properties, so it can be concluded that changing the Lehmer mean calculation procedure is beneficial in many scenarios.

For an additional set of experiments, one of the best state-of-the-art LSHADE versions, namely LSHADE-RSP (Stanovov et al., 2018) has been chosen. LSHADE-RSP was ranked 2nd in CEC 2018 competition on bound-constrained numerical optimization, and the best among DE variants participated. LSHADE-RSP used rank-based selective pressure, and a number of parameter adaptations taken from the jSO algorithm presented in (Brest et al., 2017). LSHADE-RSP was modified to have the same changing p_F parameter as described above, resulting in LSHADE-RSP-p_F algorithm.

Although for 10D and 30D the LSHADE-RSP-p_F does not show any significant difference in performance, for larger dimensions the change in mean calculation leads to several significant improvements. For 50D, only for f7 there was a performance loss, while for 7 other functions improvements were found. For 100D, the convergence of could be slower on simpler problems, such as f1-f3, however, more complex problems are solved better, with up to 13 significant improvements from 30. The graphs demonstrating the convergence process for all functions and dimensions 50 and 100 are presented in the appendix. From these graphs it can be seen that LSHADE-RSP-p_F converges slower due to larger F values and higher exploration capabilities, however eventually it gets better solutions.

Table 1: Mann-Whitney Statistical test results for LSHADE and LSHADE-p_F.

Func	D=10	D=30	D=50	D=100
f1	0	0	1	1
f2	0	0	1	1
f3	0	0	0	-1
f4	0	0	0	1
f5	0	1	1	1
f6	0	0	1	1
f7	1	0	1	-1
f8	0	1	1	1
f9	0	0	0	0
f10	0	0	0	-1
f11	0	1	1	1
f12	0	1	1	1
f13	0	0	0	1
f14	0	0	0	1
f15	0	0	1	1
f16	0	0	0	-1
f17	0	0	0	0
f18	0	0	1	1
f19	0	0	1	1
f20	0	0	-1	-1
f21	0	0	1	1
f22	0	0	0	-1
f23	1	0	1	1
f24	0	0	0	1
f25	0	0	1	1
f26	0	0	1	1
f27	0	0	1	1
f28	0	0	1	0
f29	0	0	1	0
f30	0	0	0	1
Total	1	4	17	14

5 CONCLUSIONS

In is paper the generalized Lehmer mean was proposed for calculation of control parameters in Differential Evolution adaptation process. The generalized mean formulation allows different types of mean calculation to be presented in a single equation with one control parameter. The performed experiments have shown that using larger p parameter in Lehmer

Table 2: Mann-Whitney Statistical test results for LSHADE-RSP and LSHADE-RSP-p_F.

Func	D=10	D=30	D=50	D=100
f1	0	0	0	-1
f2	0	0	0	-1
f3	0	0	0	-1
f4	0	0	0	0
f5	0	0	0	0
f6	0	0	0	1
f7	0	0	-1	0
f8	0	0	0	0
f9	0	0	0	0
f10	0	0	0	-1
f11	0	0	1	1
f12	0	0	1	1
f13	0	0	0	1
f14	0	0	0	1
f15	0	0	1	1
f16	0	0	0	0
f17	0	0	0	0
f18	0	0	1	1
f19	0	0	1	1
f20	0	0	0	0
f21	0	0	0	0
f22	0	0	0	-1
f23	0	0	0	1
f24	0	0	0	1
f25	0	0	0	0
f26	0	0	1	1
f27	0	0	1	0
f28	0	0	0	1
f29	0	0	0	0
f30	0	0	0	1
Total	0	0	6	8

mean leads to an improvement in the search properties of DE, especially for larger dimensions. For the LSHADE algorithm there were up to 17 significant improvements according to Mann-Whitney statistical tests for 50D and up to 14 improvements for 100D. It was shown that a simple heuristic rule which increases the p parameter for scaling factor F calculation with the growth of the problem dimension may improve even some of the best state-of-the-art DE variants, like LSHADE-RSP. Although the presented generalization of Lehmer mean was shown be efficient, some other mean formulation could be considered and tested.

REFERENCES

Al-Dabbagh, R. D., Neri, F., Idris, N., and Baba, M. S. (2018). Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy. In *Swarm and Evolutionary Computation 43, pp. 284–311*.

Brest, J., Maučec, M., and Boškovic, B. (2017). Single objective real-parameter optimization algorithm jSO. In *Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1311–1318*. IEEE Press.

Bullen, P. (2003). *Handbook of Means and Their Inequalities*. Springer Netherlands.

Caraffini, F., Kononova, A. V., and Corne, D. (2019). Infeasibility and structural bias in differential evolution. *Information Sciences*, 496:161 – 179.

Das, S., Mullick, S., and Suganthan, P. (2016). Recent advances in differential evolution – an updated survey. In *Swarm and Evolutionary Computation, Vol. 27, pp. 1–30*.

Peng, F., Tang, K., Chen, G., and Yao, X. (2009). Multistart jade with knowledge transfer for numerical optimization. In *2009 IEEE Congress on Evolutionary Computation, pp. 1889–1895*.

Stanovov, V., Akhmedova, S., and Semenkin, E. (2018). Lshade algorithm with rank-based selective pressure strategy for solving cec 2017 benchmark problems. *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. In *Journal of Global Optimization, Vol. 11, N. 4, pp. 341-359*.

Tanabe, R. and Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. In *IEEE Congress on Evolutionary Computation, pp. 71–78*.

Tanabe, R. and Fukunaga, A. (2014). Improving the search performance of SHADE using linear population size reduction. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC, pp. 1658–1665*.

Wu, G., R., M., and Suganthan, P. N. (2016). Problem definitions and evaluation criteria for the cec 2017 competition and special session on constrained single objective real-parameter optimization. In *Tech. rep.*

Zhang, J. and Sanderson, A. C. (2009). Jade: Adaptive differential evolution with optional external archive. In *IEEE Transactions on Evolutionary Computation 13, pp. 945–958*.

APPENDIX

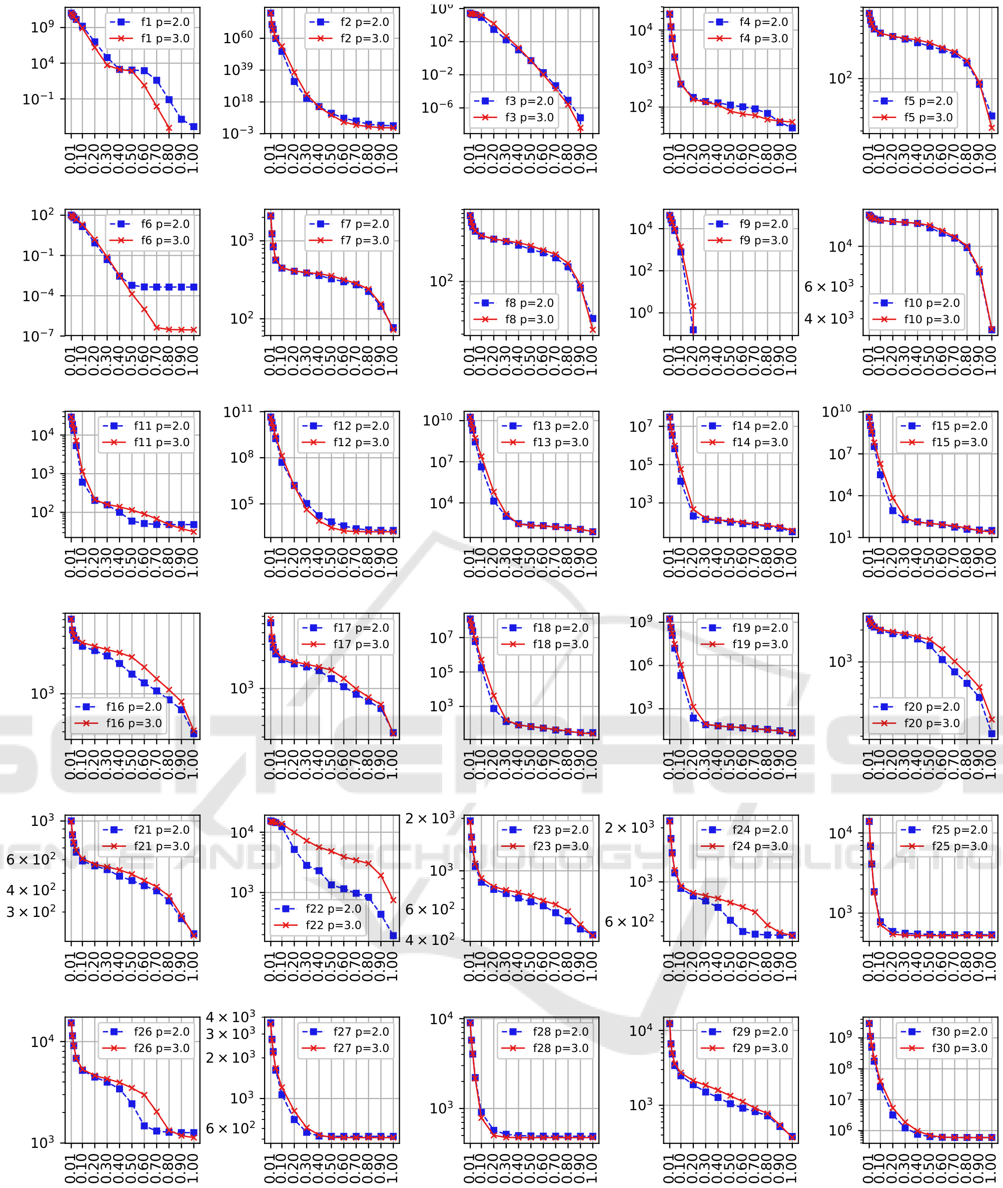


Figure 8: Convergence of LSHADE-RSP and LSHADE-RSP-p_F, D=50.

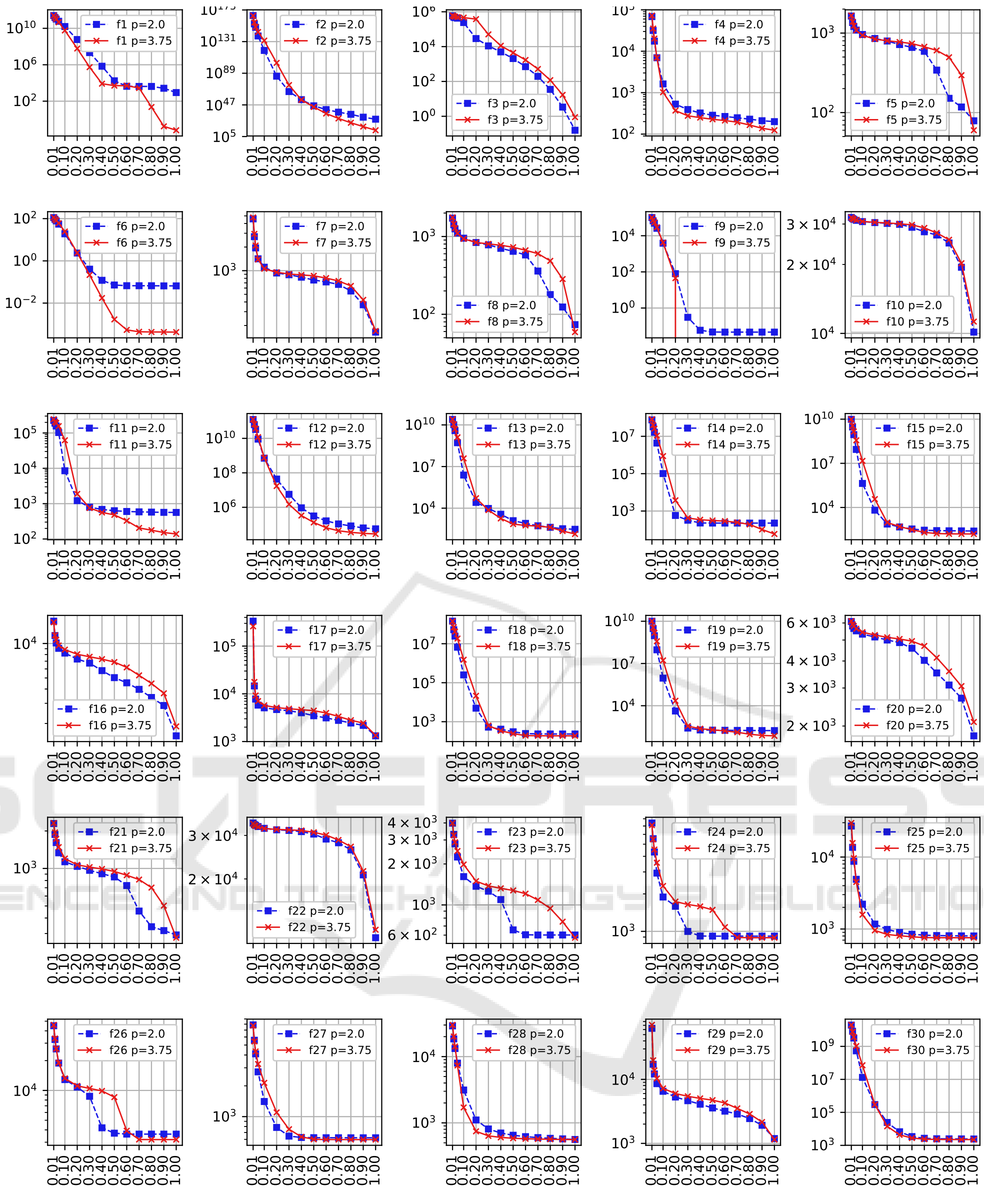


Figure 9: Convergence of LSHADE-RSP and LSHADE-RSP-p_F, D=100.