Adaptive Constraint Handling and Success History Differential Evolution for CEC 2017 Constrained Real-Parameter Optimization

Aleš Zamuda
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Smetanova ul. 17, 2000 Maribor, Slovenia
Email: ales.zamuda@um.si

***Abstract*—This paper presents Success-History Based Adaptive Differential Evolution Algorithm (SHADE) including Linear population size reduction (L-SHADE), enhanced with adaptive constraint violation handling, applied to the benchmark for CEC 2017 Competition on Constrained Real-Parameter Optimization. The constraint handling method prioritizes the feasible solutions before infeasible, while disregarding the constraint violation values below an adaptive threshold, i.e. adaptive ϵ-constraint handling. The 28 constrained test functions on 10, 30, 50, and 100 dimensions are assessed on the benchmark and the required resulting final fitnesses, constraints violations, and success rates are reported for 25 independent runs of the proposed algorithm under the budget of fixed maximum number of fitness evaluations for 10, 30, 50, and 100 dimensional test functions.**

I. Introduction

This paper presents Success-History Based Adaptive Differential Evolution Algorithm (SHADE [1]) with linear population size reduction (L-SHADE [2]), enhanced with constraint handling [3], [4], applied to CEC 2017 Competition on Constrained Real-Parameter Optimization [5].

The applied constraint handling method prioritizes the feasible solutions before infeasible, while disregarding the constraint violation values below an adaptive threshold, i.e. adaptive ϵ-constraint handling [6], [3], [4]. The 28 constrained test functions on 10, 30, 50, and 100 dimensions are assessed on the CEC benchmark and the resulting final fitnesses, constraints violations, and success rates are reported for 25 independent runs of the proposed algorithm under the budget of fixed maximum number of fitness evaluations.

In the next section, related work is presented. The description of the proposed constrained L-SHADE follows in Section 3. Section 4 reports experiments and Section 5 conclusions with future work.

II. Related Work

A. Differential Evolution and Optimization

Differential Evolution (DE) was introduced by Storn and Price [7] using a floating-point encoding evolutionary algorithm [8] for global optimization over continuous spaces. There exist several recent surveys and insights with the algorithm's base name [9], [10], [11], [12], [13], [14] and its metaphors [15], [16], [17] stemming from the progress on computational mechanisms mainly from the branches of DE.

The basic DE [7] consists of an evolutionary loop over generations, within which new population D-dimensional vectors $\mathbf{x}_i$, $\forall i \in \{1, 2, ..., NP\}$ are evolved. During each generation step number g, computational operators like mutation, crossover, and selection on the population are performed, until a termination criterion is satisfied, like a fixed number of maximum fitness evaluations (MAXFES). DE and recently the L-SHADE [2] as applied in this paper, has also been extended with several enhancements for unconstrained optimization, like [18], [19], [20], [21], [22], [23].

DE is also used for constraint optimization [24] and there are several variants which entered past competitions held at the IEEE Congress on Evolutionary Computation (CEC) [25]. A comparison of several constraint-handling techniques entering these competitions was recently demonstrated in [4] on an uniform framework of limited fitness evaluation budget for underwater robotics challenge in robust glider path planning, showcasing some well known algorithms, like SaDE [26], jDE and SPSRDEMMS extensions [27], [3], [12], JADE [28], ECHT-DE and applications [29], [30], [31], [32], EPSDE [33], CoDE [34], DMS-PSO [35], CLPSO [36], CMAES [37], ABC [38], and MABC [39]. The optimization algorithms within these 11 algorithmic metaphor expressions were also enhanced with a same constraint handling approach, which also improved the performance for some of their incorporated constraint handling techniques, like the one in MABC. A surrogate matrix DE with an ϵ-constraint handling method was also recently applied in scheduling hydro and thermal power systems production [3], building upon ϵ-constraint method with handling of the adjustable ϵ error level [6], presented on multi-objective optimization functions.

B. Constrained Optimization

With regard to inequality constraints $g_i(\mathbf{x}) \leq \mathbf{0}$ and equality constraints $h_j(\mathbf{x}) = \mathbf{0}$, a solution $\mathbf{x}$ is regarded as *feasible* if

$$g_i(\mathbf{x}) \leq 0, \quad i = \quad 1, ..., q, \tag{1}$$

$$|h_j(\mathbf{x})| - \epsilon \leq 0, \quad j = \quad q+1, ..., m, \tag{2}$$

978-1-5090-4601-0/17/$31.00

Table I

FUNCTION VALUES ACHIEVED WHEN FES = 2×10^5 FOR $10D$ PROBLEMS C01 – C06.

FEs		C01	C02	C03	C04	C05	C06
2×10^5	Best	0.000000	0.000000	6341.810292	15.919244	0.000000	103.288465
	Median	0.000000	0.000000	40103.199303	35.818324	0.000000	307.643490
	c	0, 0, 0	0, 0, 0	0, 0, 1	0, 0, 0	0, 0, 0	0, 0, 5
	$\overline{\nu}$	0.000000	0.000000	0.000103	0.000000	0.000000	0.000000
	Mean	0	0	110008	38.738	0.956779	549.617
	Worst	0.000000	0.000000	548034.199888	55.717399	3.986579	2058.812018
	STD	0.0000e+00	0.0000e+00	1.5587e+05	8.9484e+00	1.7377e+00	4.8668e+02
	SR	100%	100%	44%	100%	100%	96%
	$\overline{vio}$	0	0	0.00063352	0	0	0.0053656

Table II

FUNCTION VALUES ACHIEVED WHEN FES = 2×10^5 FOR $10D$ PROBLEMS C07 – C12.

FEs		C07	C08	C09	C10	C11	C12
2×10^5	Best	-148.219878	-0.001348	-0.004975	-0.000510	-0.168819	3.987902
	Median	-65.209283	-0.001348	-0.004975	-0.000510	-0.168819	3.987902
	c	0, 0, 2	0, 0, 2	0, 0, 1	0, 0, 1	0, 0, 1	0, 0, 0
	$\overline{\nu}$	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
	Mean	-48.7352	-0.001348	0.125471	-0.00051	-0.156266	3.9879
	Worst	102.366112	-0.001348	3.256178	-0.000510	-0.092195	3.987902
	STD	6.8267e+01	6.6394e-19	6.5223e-01	0.0000e+00	2.4170e-02	9.0649e-16
	SR	68%	100%	100%	100%	100%	100%
	$\overline{vio}$	0.00309144	1.456e-05	4e-06	3.96e-06	2e-05	0

Table III

FUNCTION VALUES ACHIEVED WHEN FES = 2×10^5 FOR $10D$ PROBLEMS C13 – C18.

FEs		C13	C14	C15	C16	C17	C18
2×10^5	Best	0.000000	2.376332	8.808759	38.920288	0.735418	216.729460
	Median	0.000000	2.628816	14.922494	64.402746	1.009918	704.500000
	c	0, 0, 0	0, 0, 1	0, 0, 1	0, 0, 1	1, 0, 1	1, 0, 0
	$\overline{\nu}$	0.000000	0.000000	0.000000	0.000051	5.501185	971839.452774
	Mean	1.11624	2.64069	16.2454	65.4166	1.00008	1348.61
	Worst	3.986579	3.289782	27.490020	89.535116	1.099044	8598.750000
	STD	1.8269e+00	2.3734e-01	3.8242e+00	1.3705e+01	6.7542e-02	1.7324e+03
	SR	100%	100%	68%	60%	0%	0%
	$\overline{vio}$	0	0	15.0705	0.0129993	5.5448	1.40366e+08

Table IV

FUNCTION VALUES ACHIEVED WHEN FES = 2×10^5 FOR $10D$ PROBLEMS C19 – C24.

FEs		C19	C20	C21	C22	C23	C24
2×10^5	Best	0.000001	0.509405	3.987902	0.000000	2.376335	8.639310
	Median	0.000003	0.757646	3.987902	0.000000	2.628819	14.922494
	c	1, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 1	0, 0, 1
	$\overline{\nu}$	4422.394773	0.000000	0.000000	0.000000	0.000000	0.004680
	Mean	3.44e-06	0.868614	6.86278	6.6684	2.52785	15.7361
	Worst	0.000005	1.459877	22.785292	88.330945	2.633666	24.353961
	STD	8.2057e-07	2.3874e-01	5.4394e+00	2.1067e+01	1.2624e-01	3.3427e+00
	SR	0%	100%	100%	100%	100%	40%
	$\overline{vio}$	4422.39	0	0	0	1.2e-05	0.100907

Table V
FUNCTION VALUES ACHIEVED WHEN FES = 2×10^5 FOR $10D$ PROBLEMS C25 – C28.

FEs		C25	C26	C27	C28
2×10^5	Best	43.980493	0.914313	295.741099	0.000003
	Median	62.831714	1.008974	1020.389563	31.511898
	c	0, 0, 1	1, 0, 1	1, 0, 0	1, 0, 0
	$\overline{\nu}$	0.000000	5.500259	1959116.576504	4438.936128
	Mean	61.1432	1.00191	2438.8	31.1262
	Worst	76.968881	1.061495	18140.750000	59.728169
	STD	9.5260e+00	3.0885e-02	3.7684e+03	1.2317e+01
	SR	60%	0%	0%	0%
	$\overline{vio}$	0.012243	5.50499	1.26381e+07	4436.25

Table VI
FUNCTION VALUES ACHIEVED WHEN FES = 6×10^5 FOR $30D$ PROBLEMS C01 – C06.

FEs		C01	C02	C03	C04	C05	C06
6×10^5	Best	0.000000	0.000000	217854.405028	64.671883	0.000000	1976.358211
	Median	0.000000	0.000000	736404.820848	113.424634	0.000000	3827.588288
	c	0, 0, 0	0, 0, 0	0, 0, 1	0, 0, 0	0, 0, 0	0, 0, 4
	$\overline{\nu}$	0.000000	0.000000	0.001441	0.000000	0.000000	0.000000
	Mean	0	0	1.29926e+06	115.734	0.797325	3745.32
	Worst	0.000000	0.000000	5082420.837959	159.192594	3.986624	5065.298248
	STD	0.0000e+00	0.0000e+00	1.1954e+06	2.2016e+01	1.6275e+00	8.4312e+02
	SR	100%	100%	32%	100%	100%	100%
	$\overline{vio}$	0	0	0.0242756	0	0	1.164e-05

Table VII
FUNCTION VALUES ACHIEVED WHEN FES = 6×10^5 FOR $30D$ PROBLEMS C07 – C12.

FEs		C07	C08	C09	C10	C11	C12
6×10^5	Best	-330.786337	-0.000284	-0.002666	-0.000103	-14.667088	9.775008
	Median	-32.589365	-0.000284	-0.002666	-0.000103	-0.919556	9.775177
	c	0, 0, 2	0, 0, 2	0, 0, 1	0, 0, 1	0, 0, 1	0, 0, 0
	$\overline{\nu}$	0.000067	0.000000	0.000000	0.000000	0.000000	0.000000
	Mean	-24.1162	-0.000284	0.0233628	-0.000103	0.780843	14.2274
	Worst	185.582813	-0.000284	0.648053	-0.000103	14.600080	28.326170
	STD	1.1546e+02	1.6598e-19	1.3014e-01	4.1496e-20	7.7193e+00	8.0862e+00
	SR	52%	100%	96%	100%	100%	100%
	$\overline{vio}$	0.0035614	0	1.07092e+06	6.6e-06	2.4e-05	1.6e-07

Table VIII
FUNCTION VALUES ACHIEVED WHEN FES = 6×10^5 FOR $30D$ PROBLEMS C13 – C18.

FEs		C13	C14	C15	C16	C17	C18
6×10^5	Best	0.000000	1.408518	18.064087	150.753290	1.026510	542.988642
	Median	145.149023	1.495440	23.638932	221.482144	1.029992	1500.362863
	c	0, 0, 0	0, 0, 1	0, 0, 1	0, 0, 1	1, 0, 1	1, 0, 0
	$\overline{\nu}$	0.000000	0.000000	0.000000	0.002821	15.500531	767708.370241
	Mean	3534.75	1.54841	24.0546	211.879	1.03123	6705.49
	Worst	81245.562930	2.230469	35.400797	238.896901	1.051973	126317.342201
	STD	1.6194e+04	2.4856e-01	4.4986e+00	2.7059e+01	4.8130e-03	2.4939e+04
	SR	100%	100%	48%	16%	0%	0%
	$\overline{vio}$	4e-08	2e-06	201.143	0.00883192	17.306	8.75956e+08

Table IX
FUNCTION VALUES ACHIEVED WHEN FES = 6×10^5 FOR $30D$ PROBLEMS C19 – C24.

FEs		C19	C20	C21	C22	C23	C24
6×10^5	Best	0.000007	1.489221	3.982525	80.601739	1.430648	11.774784
	Median	0.000010	2.077242	9.775200	17993.997690	1.492347	18.820565
	c	1, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 1, 0
	$\overline{\nu}$	14249.938713	0.000000	0.000000	0.000000	0.000000	0.043439
	Mean	0.309415	2.11336	13.262	34214.7	1.58179	20.8766
	Worst	7.735134	3.252612	28.326226	170135.261783	2.237525	32.873343
	STD	1.5470e+00	3.5057e-01	9.0184e+00	4.1932e+04	2.3652e-01	4.2899e+00
	SR	0%	100%	100%	72%	84%	20%
	$\overline{vio}$	14250.2	0	4e-08	0.275674	0.00016036	14.7372

Table X
FUNCTION VALUES ACHIEVED WHEN FES = 6×10^5 FOR $30D$ PROBLEMS C25 – C28.

FEs		C25	C26	C27	C28
6×10^5	Best	164.932535	1.020516	771.693355	70.679844
	Median	213.628161	1.029911	2328.966746	129.800433
	c	0, 0, 1	1, 0, 1	1, 0, 0	1, 0, 0
	$\overline{\nu}$	0.000966	15.503185	4086122.896608	14309.936442
	Mean	207.224	1.03526	3141.71	122.421
	Worst	238.758353	1.103004	9025.250000	167.997621
	STD	2.1262e+01	1.7386e-02	2.0057e+03	2.9127e+01
	SR	32%	0%	0%	0%
	$\overline{vio}$	0.0233978	21.5051	1.51914e+07	14311.2

Table XI
FUNCTION VALUES ACHIEVED WHEN FES = 1×10^6 FOR $50D$ PROBLEMS C01 – C06.

FEs		C01	C02	C03	C04	C05	C06
1×10^6	Best	0.000000	0.000000	460407.836440	145.263065	0.000000	3486.644298
	Median	0.000000	0.000000	4381259.215675	181.081674	0.000000	6041.018996
	c	0, 0, 0	0, 0, 0	0, 0, 1	0, 0, 0	0, 0, 0	0, 0, 4
	$\overline{\nu}$	0.000000	0.000000	0.000050	0.000000	0.000000	0.000000
	Mean	0	0	6.64133e+06	187.37	0.31893	6364.72
	Worst	0.000000	0.000000	27234258.492770	244.758532	3.986624	9005.415965
	STD	0.0000e+00	0.0000e+00	5.9790e+06	2.5905e+01	1.1038e+00	1.6322e+03
	SR	100%	100%	48%	100%	100%	100%
	$\overline{vio}$	0	0	0.0694317	0	0	1.02e-05

Table XII
FUNCTION VALUES ACHIEVED WHEN FES = 1×10^6 FOR $50D$ PROBLEMS C07 – C12.

FEs		C07	C08	C09	C10	C11	C12
1×10^6	Best	-340.224874	0.000601	-0.002037	-0.000047	-109.421403	7.068349
	Median	-85.989214	0.000965	-0.002037	-0.000045	-7.725566	17.938526
	c	0, 0, 2	0, 0, 0	0, 0, 1	0, 0, 0	0, 0, 1	0, 0, 0
	$\overline{\nu}$	0.000075	0.000000	0.000000	0.000000	0.000768	0.000000
	Mean	-68.1059	0.0009928	0.0810008	-4.284e-05	4.75824	24.76
	Worst	163.958553	0.001558	1.138593	-0.000018	174.974166	36.406623
	STD	1.3458e+02	2.4328e-04	2.3626e-01	6.1011e-06	6.2138e+01	1.1073e+01
	SR	56%	100%	84%	100%	20%	100%
	$\overline{vio}$	0.00180008	3.48e-06	1.47087e+07	0	0.0373406	0

Table XIII
FUNCTION VALUES ACHIEVED WHEN FES = 1×10^6 FOR $50D$ PROBLEMS C13 – C18.

FEs		C13	C14	C15	C16	C17	C18
1×10^6	Best	1167.328018	1.099952	16.253533	252.899798	1.044582	1920.854153
	Median	19138.956725	1.152444	30.432831	334.568619	1.049711	2532.768976
	c	0, 0, 0	0, 0, 1	2, 0, 0	0, 0, 1	1, 0, 1	2, 0, 0
	$\overline{\nu}$	0.000000	0.000000	1111.291761	0.003569	25.501255	7205925.387777
	Mean	28658.5	1.22882	30.4855	347.032	1.05454	3195.71
	Worst	124972.436824	1.693323	44.803744	409.974301	1.112135	8811.250000
	STD	2.7747e+04	2.0999e-01	7.5576e+00	4.0283e+01	1.7061e-02	1.5510e+03
	SR	88%	100%	12%	12%	0%	0%
	$\overline{vio}$	0.275684	6e-06	1465.15	0.0170548	35.1776	2.48961e+07

Table XIV
FUNCTION VALUES ACHIEVED WHEN FES = 1×10^6 FOR $50D$ PROBLEMS C19 – C24.

FEs		C19	C20	C21	C22	C23	C24
1×10^6	Best	0.000019	2.793541	3.981450	15066.307937	1.109489	18.062034
	Median	0.000023	3.586492	17.938513	51633.342274	1.124193	20.491141
	c	1, 0, 0	0, 0, 0	0, 0, 0	1, 0, 0	0, 0, 1	0, 1, 0
	$\overline{\nu}$	24077.482653	0.000000	0.000000	0.272316	0.000000	0.016543
	Mean	0.301994	3.61902	12.6618	61762.5	1.14533	20.9023
	Worst	7.549308	4.399405	17.938649	161513.848015	1.650535	28.006418
	STD	1.5099e+00	3.7602e-01	6.1584e+00	3.9523e+04	1.0587e-01	2.9648e+00
	SR	0%	100%	100%	36%	84%	28%
	$\overline{vio}$	24077.7	5.2e-07	0	3.47373	9.384e-05	1.24862

Table XV
FUNCTION VALUES ACHIEVED WHEN FES = 1×10^6 FOR $50D$ PROBLEMS C25 – C28.

FEs		C25	C26	C27	C28
1×10^6	Best	290.597196	1.047177	3165.877342	161.591600
	Median	347.144603	1.049756	5995.699913	207.781913
	c	0, 0, 1	1, 0, 1	1, 0, 0	1, 0, 0
	$\overline{\nu}$	0.000741	25.500075	30744620.165203	24187.293211
	Mean	353.965	1.04949	7159.79	217.154
	Worst	414.690091	1.051245	15848.373852	278.318312
	STD	3.5452e+01	1.0639e-03	3.0852e+03	3.1720e+01
	SR	32%	0%	0%	0%
	$\overline{vio}$	0.0165844	25.5381	4.81677e+07	24191.2

Table XVI
FUNCTION VALUES ACHIEVED WHEN FES = 2×10^5 FOR $100D$ PROBLEMS C01 – C06.

FEs		C01	C02	C03	C04	C05	C06
2×10^6	Best	0.080255	0.072938	1684503.319181	329.329439	0.000000	10950.209682
	Median	0.432564	0.184568	9938948.898056	408.925707	0.011586	15506.558185
	c	0, 0, 0	0, 0, 0	0, 0, 1	0, 0, 0	0, 0, 0	0, 0, 2
	$\overline{\nu}$	0.000000	0.000000	0.002547	0.000000	0.000000	0.000000
	Mean	0.977746	0.366104	1.51413e+07	413.582	0.818836	15222.9
	Worst	11.315168	3.620979	60598481.747409	469.617973	4.066555	18535.330298
	STD	2.1781e+00	6.9971e-01	1.3449e+07	3.6721e+01	1.5122e+00	1.7824e+03
	SR	100%	100%	16%	100%	100%	100%
	$\overline{vio}$	0	0	0.0242065	0	0	5.6e-06

Table XVII
FUNCTION VALUES ACHIEVED WHEN FES = 2×10^6 FOR $100D$ PROBLEMS C07 – C12.

FEs		C07	C08	C09	C10	C11	C12
2×10^6	Best	-481.328981	0.013288	0.000000	0.000365	-311.381389	9.995419
	Median	-278.650432	0.027209	0.000217	0.000501	-50.360027	18.857919
	c	0, 0, 2	0, 0, 2	0, 0, 0	0, 0, 0	0, 1, 0	0, 0, 0
	$\overline{\nu}$	0.000198	0.000832	0.000000	0.000000	0.101550	0.000000
	Mean	-193.458	0.0415975	0.522499	0.00051308	-13.7384	23.8996
	Worst	376.526002	0.087460	5.348516	0.000684	270.480165	31.577401
	STD	2.0127e+02	2.4668e-02	1.1223e+00	7.3482e-05	1.6005e+02	8.0143e+00
	SR	40%	0%	96%	100%	0%	100%
	$\overline{vio}$	0.00436664	0.0012406	1.564e-05	1.268e-05	0.14709	0

Table XVIII
FUNCTION VALUES ACHIEVED WHEN FES = 2×10^6 FOR $100D$ PROBLEMS C13 – C18.

FEs		C13	C14	C15	C16	C17	C18
2×10^6	Best	47240.433308	0.784202	21.205680	603.274296	1.081255	8014.814662
	Median	105873.476475	0.784209	27.764485	697.443169	1.099890	9546.835255
	c	1, 0, 0	0, 0, 1	0, 0, 1	0, 1, 0	1, 0, 1	1, 0, 0
	$\overline{\nu}$	16.788208	0.000000	0.000000	0.005923	50.502033	76376469.238959
	Mean	118844	0.79492	30.8846	712.859	1.09805	16640.7
	Worst	351749.503529	0.840236	54.258518	823.112306	1.101082	174544.457041
	STD	6.3069e+04	1.5593e-02	8.4372e+00	5.5495e+01	5.1411e-03	3.2936e+04
	SR	0%	100%	60%	12%	0%	0%
	$\overline{vio}$	18.7453	2e-05	1253.21	0.0488209	54.2153	4.05209e+09

Table XIX
FUNCTION VALUES ACHIEVED WHEN FES = 2×10^6 FOR $100D$ PROBLEMS C19 – C24.

FEs		C19	C20	C21	C22	C23	C24
2×10^6	Best	0.000066	6.311732	3.980643	122274.131508	0.800967	18.097334
	Median	0.000088	7.342300	9.995811	259925.383174	0.810964	21.817367
	c	1, 0, 0	0, 0, 0	0, 0, 0	1, 0, 0	0, 0, 1	0, 1, 0
	$\overline{\nu}$	48646.342504	0.000000	0.000000	46.885568	0.000000	0.323019
	Mean	8.82e-05	7.40385	14.9419	263043	0.814437	22.8996
	Worst	0.000122	8.571167	31.578069	508308.160038	0.844395	31.156972
	STD	1.3235e-05	5.5579e-01	7.3058e+00	9.7856e+04	1.0029e-02	2.8828e+00
	SR	0%	100%	100%	0%	92%	8%
	$\overline{vio}$	48646.3	2e-07	3.2e-07	46.1771	3.684e-05	234.413

Table XX
FUNCTION VALUES ACHIEVED WHEN FES = 2×10^6 FOR $100D$ PROBLEMS C25 – C28.

FEs		C25	C26	C27	C28
2×10^6	Best	642.455558	1.097472	28172.148693	354.855690
	Median	717.802821	1.099953	47450.441847	410.517414
	c	0, 0, 1	1, 0, 1	2, 0, 0	1, 0, 0
	$\overline{\nu}$	0.004779	50.500583	1106046245.109879	48875.776527
	Mean	724.212	1.10092	47247.7	415.526
	Worst	797.951458	1.125744	84021.726414	492.948144
	STD	3.2369e+01	5.3132e-03	1.2565e+04	3.5785e+01
	SR	24%	0%	0%	0%
	$\overline{vio}$	0.0295336	52.6093	1.37064e+09	48880.7

where equality constraints are transformed into inequalities. A mean value of all constraints' violations $\overline{\nu}$ is defined as:

$$\overline{\nu} = \frac{(\sum_{i=1}^{q} G_i(\mathbf{x}) + \sum_{j=q+1}^{m} H_j(\mathbf{x}))}{m}, \quad (3)$$

where the sum of all constraint violations is zero for feasible solutions and positive when at least one constraint is violated:

$$G_i(\mathbf{x}) = \begin{cases} g_i(\mathbf{x}), & g_i(\mathbf{x}) > 0, \\ 0, & g_i(\mathbf{x}) \leq 0, \end{cases} \quad (4)$$

$$H_j(\mathbf{x}) = \begin{cases} |h_j(\mathbf{x})|, & |h_j(\mathbf{x})| - \epsilon > 0, \\ 0, & |h_j(\mathbf{x})| - \epsilon \leq 0. \end{cases} \quad (5)$$

To compare two solutions during the selection operation, a new vector $\mathbf{x}_{i,g+1}$ can be produced based on the adapted error level of the mean constraint violation $\overline{\nu}$:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{x}_{j,g} & \text{if } (\overline{\nu}_{i,g} > \overline{\nu}_{j,g}), \\ \mathbf{x}_{j,g} & \text{else if } (\overline{\nu}_{j,g} = 0) \wedge (f(\mathbf{x}_{i,g}) > f(\mathbf{x}_{j,g})), \\ \mathbf{x}_{i,g} & \text{otherwise}, \end{cases} \quad (6)$$

where ϵ level can be controlled, e.g. by the mechanism as defined by Takahama and Sakai in [40]. The ϵ level can be updated until the number of generations g reaches the control generation g_c and when the number of generations exceeds g_c, the ϵ level can be set at 0 to obtain solutions with minimum constraint violations, as applied in [41], [6], [42], [3], [4].

III. Constraint Handling with Success History Adaptive Differential Evolution for CEC 2017 Constrained Real-Parameter Optimization (CAL-SHADE)

The constrained L-SHADE, CAL-SHADE, uses constraints and adaptation of epsilon value, i.e. epsilon level handling to adapt constraints comparisons. To enhance L-SHADE with this adaptive constraint handling, the equations (1)–(5) are used for constraints violation and aggregation computation. Then, in the individual vectors comparisons, the equation (6) is used. When computing difference in success history adaptation and when there are constraint violation improvements, the constraints take precedence.

CAL-SHADE uses an initial population size of $NP_{\text{init}} = 2 \times D$, p value for current-to-$p$best/1 mutation 0.11, historical memory size of $H = 5$, and external archive size $|A|$ of N_{init} multiplied by $r^{\text{arc}} = 2$. The initial ϵ level is set at Deb-rules ranked individual at 0.2 NP-th individual, and at $\theta_g = 0.8NP$-th in later generations; ϵ level is diminished to order of 5, and after $g_c = 500$ generations this ϵ level relaxation is omitted to fully consider all the constraints [6]. Except the NP_{init}, θ_g, H, and r^{arc}, the parameter settings are therefore taken from the literature [4], [2].

IV. Experiments

According to [5], MAXFES was set at 20000*D and 25 independent runs of the proposed algorithm were recorded on 28 test problem functions for $\forall D \in \{10, 30, 50, 100\}$, with all other parameters same as in the original algorithms. The results are listed in Tables I–XX, where according to [5], c is the number of violated constraints at the median solution (the sequence of three numbers indicate the number of violations — including inequalities and equalities — by more than 1.0, in the range [0.01, 1.0], and in the range [0.0001, 0.01], respectively; the count can be non-zero for a feasible solution; note $q_{\text{C9}} = 1$), $\overline{\nu}$ is the mean value of violations of all constraints at the median solution, SR is the feasibility rate of the solutions obtained in 25 runs, and $\overline{vio}$ is the mean constraint violation value of all the solutions of 25 runs.

Table XXI
Parameter tuning for CAL-SHADE and the count of 100% success rate functions (100% SR – solved problems).

Setting No.	NP_{init}	H	r^{arc}	θ_g	SR
1	2	5	1.4	80%	**28**
2	1	5	1.4	20%	26
3	1	6	2.6	20%	22
4	1	6	2.6	80%	21
6	2	5	2	20%	23
7	2	6	2.6	20%	22
8	2	6	2.6	80%	22
9	2	5	1.4	20%	26
10	3	5	1.4	20%	23
11	4	5	1.4	20%	22
12	5	5	1.4	20%	21
13	5	6	2.6	20%	18

The PC Configure is as follows. System: "Ubuntu 16.04.1 LTS". CPU: "Intel(R) Core(TM) i5-3550 CPU @ 3.30GHz". RAM: 8GB. Language: Slovene and English. Algorithm: CAL-SHADE. With Matlab implementation of this algorithm, its complexity as defined in [5], is measured on 10000 FES as: T1 is 0.0781083 s, T2 is 74.27 s, (T2-T1)/T1 is 949.86.

Table XXI includes tuning experiments for the algorithm, with 14 different settings reported. The last one is most similar to the original L-SHADE, while the best ranked according to number of solved problems is the top one, the CAL-SHADE.

V. Conclusion

This paper presented L-SHADE, enhanced with constraint handling, applied to CEC 2017 functions. The constraint handling method prioritizes the feasible solutions before infeasible, while disregarding the constraint violation values below an adaptive threshold, i.e. adaptive ϵ constraint handling. The 28 constrained test functions on 10, 30, 50, and 100 dimensions were assessed on the benchmark and the resulting final fitnesses, constraints violations, and success rates were reported for 25 independent runs of the proposed algorithm under the budget of fixed maximum number of fitness evaluations. Future work includes testing other constraint handling approaches within this framework as well as testing other mechanisms and applications to other domains.

Acknowledgment

This work was supported by the Slovenian Research Agency under project P2-0041 and COST Action CA15140: Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO). Part of codes in Matlab for extending the optimization algorithms utilized are provided by Qingfu Zhang at http://dces.essex.ac.uk/staff/qzhang/code/ and by Ryoji Tanabe's Home Page at https://sites.google.com/site/tanaberyoji/home.

References

[1] R. Tanabe and A. Fukunaga, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2013, pp. 1952–1959.

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

[3] A. Glotić and A. Zamuda, "Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution," *Applied Energy*, vol. 141, pp. 42–56, 1 March 2015.

[4] A. Zamuda, J. D. H. Sosa, and L. Adler, "Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Sub-mesoscale Eddy Sampling," *Applied Soft Computing*, vol. 42, pp. 93–118, 2016.

[5] G. Wu, R. Mallipedi, and S. P. N., "Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization," National University of Defense Technology, Changsha, Hunan, P.R. China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report 201709, 2017.

[6] A. Zamuda, J. Brest, B. Bošković, and V. Žumer, "Differential Evolution with Self-adaptation and Local Search for Constrained Multiobjective Optimization," in *IEEE Congress on Evolutionary Computation 2009*. IEEE Press, 2009, pp. 195–202.

[7] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[8] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003.

[9] 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.

[10] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[11] S. Das, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization – A survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71–88, 2011.

[12] A. Zamuda and J. Brest, "Self-adaptive control parameters' randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

[13] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[14] A. P. Piotrowski, "Review of differential evolution population size," *Swarm and Evolutionary Computation*, vol. 32, pp. 1–24, 2017.

[15] K. Sörensen, "Metaheuristicsthe metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.

[16] I. BoussaïD, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.

[17] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[18] R. Polakova, J. Tvrdík, and P. Bujok, "Controlled restart in differential evolution applied to CEC2014 benchmark functions," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, 2014, pp. 2230–2236.

[19] S.-M. Guo, J. S.-H. Tsai, C.-C. Yang, and P.-H. Hsu, "A self-optimization approach for l-shade incorporated with eigenvector-based crossover and successful-parent-selecting framework on cec 2015 benchmark set," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1003–1010.

[20] N. Awad, M. Z. Ali, and R. G. Reynolds, "A differential evolution algorithm with success-based parameter adaptation for cec2015 learning-based optimization," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1098–1105.

[21] R. Polakova, J. Tvrdík, and P. Bujok, "L-SHADE with Competing Strategies Applied to CEC2015 Learning-based Test Suite," in *2016 IEEE World Congress on Computational Intelligence (IEEE WCCI 2016), Vancouver, Canada*, 2016, pp. 4790–4796.

[22] N. H. Awad, M. Z. Ali, and R. R. G. Suganthan, Ponnuthurai N. and, "An Ensemble Sinusoidal Parameter Adaptation incorporated with L-SHADE for Solving CEC2014 Benchmark Problems," in *2016 IEEE World Congress on Computational Intelligence (IEEE WCCI 2016), Vancouver, Canada*, 2016, pp. 2958–2965.

[23] J. Brest, M. S. Maučec, and B. Bošković, "iL-SHADE: Improved L-SHADE Algorithm for Single Objective Real-Parameter Optimization," in *2016 IEEE World Congress on Computational Intelligence (IEEE WCCI 2016), Vancouver, Canada*, 2016, pp. 1188–1195.

[24] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[25] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., 2010.

[26] 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.

[27] J. Brest and M. S. Maučec, "Population Size Reduction for the Differential Evolution Algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.

[28] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[29] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.

[30] ——, "Differential evolution with ensemble of constraint handling techniques for solving cec 2010 benchmark problems," in *2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2010, pp. 1–8.

[31] G. Wu, W. Pedrycz, P. Suganthan, and R. Mallipeddi, "A variable reduction strategy for evolutionary algorithms handling equality constraints," *Applied Soft Computing*, vol. 37, pp. 774–786, 2015.

[32] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329–345, 2016.

[33] 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.

[34] 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, no. 1, pp. 55–66, 2011.

[35] J. J. Liang and P. Suganthan, "Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism," in *IEEE Congress on Evolutionary Computation 2006*. IEEE, 2006, pp. 9–16.

[36] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[37] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[38] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[39] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3021–3031, 2011.

[40] T. Takahama, S. Sakai, and N. Iwane, "Solving Nonlinear Constrained Optimization Problems by the ϵ Constrained Differential Evolution," *IEEE International Conference on Systems, Man and Cybernetics 2006 (SMC 2006)*, vol. 3, pp. 2322–2327, 2006.

[41] J. Brest, "Constrained Real-Parameter Optimization with ϵ-Self-Adaptive Differential Evolution," in *Constraint-Handling in Evolutionary Optimization*. Springer, 2009, pp. 73–93.

[42] J. Brest, B. Bošković, and V. Žumer, "An Improved Self-adaptive Differential Evolution Algorithm in Single Objective Constrained Real-Parameter Optimization," in *2010 IEEE World Congress on Computational Intelligence, 18. - 23. julij, Barcelona, Španija*, 2010, pp. 1073–1080.