Technical Report: An Efficient Reconstructed Differential Evolution (RDE) Variant by Some of the Current State-of-the-art Strategies for Solving Single Objective Bound Constrained Problems

1st Sichen Tao
Faculty of Engineering
University of Toyama
Toyama, Japan
taosc73@hotmail.com
d2278002@ems.u-toyama.ac.jp

2nd Ruihan Zhao
School of Mechanical Engineering
Tongji University
Shanghai, China
2110415@tongji.edu.cn

3rd Kaiyu Wang
Faculty of Engineering
University of Toyama
Toyama, Japan
greskofairy@gmail.com

4th Shangce Gao
Faculty of Engineering
University of Toyama
Toyama, Japan
gaosc@eng.u-toyama.ac.jp

***Abstract*—Complex single-objective bounded problems are often difficult to solve. In evolutionary computation methods, since the proposal of differential evolution algorithm in 1997, it has been widely studied and developed due to its simplicity and efficiency. These developments include various adaptive strategies, operator improvements, and the introduction of other search methods. After 2014, research based on LSHADE has also been widely studied by researchers. However, although recently proposed improvement strategies have shown superiority over their previous generation's first performance, adding all new strategies may not necessarily bring the strongest performance. Therefore, we recombine some effective advances based on advanced differential evolution variants in recent years and finally determine an effective combination scheme to further promote the performance of differential evolution. In this paper, we propose a strategy recombination and reconstruction differential evolution algorithm called reconstructed differential evolution (RDE) to solve single-objective bounded optimization problems. Based on the benchmark suite of the 2024 IEEE Congress on Evolutionary Computation (CEC2024), we tested RDE and several other advanced differential evolution variants. The experimental results show that RDE has superior performance in solving complex optimization problems.**

I. The Reconstructured Differential Evolution (RDE)

The main paper of RDE has published on arXiv [1]. The source code will be available at https://github.com/SichenTao. The pseudo-code of RDE is given in Algorithm 1.

1) Initialization:

$$x_{i,j} = rand(x_{L,j}, x_{U,j}) \tag{1}$$

2) Classical Mutation Strategy:

$$v_{i,j}^{(k+1)} = x_{i,j}^{(k)} + F\left(x_{p,j}^{(k)} - x_{i,j}^{(k)}\right) + F\left(x_{r1,j}^{(k)} - x_{r2,j}^{(k)}\right) \tag{2}$$

3) Classical Crossover:

$$u_{i,j}^{(k)} = \begin{cases} v_{i,j}^{(k)}, & j = j_{rand} \text{ or } rand < Cr \\ x_{i,j}^{(k)}, & \text{otherwise} \end{cases} \tag{3}$$

Algorithm 1: RDE

Input: Parameters max_nfes, D, $N_{max} = 18D$, $H = 5$, $p_{max} = 0.25$, $Ar = 1$, $\mu_F = 0.3$, $\mu_{Cr} = 0.8$, $\gamma_1 = \gamma_2 = 0.5$, $k_r = 3$ and $p_r = 0.2$
Output: The obtained best solution

```
Initialization: Randomly generate a population
  {x_1, x_2, ..., x_N} by Eq. (1), set nfes = 0
while nfes < max_nfes do
    k = k + 1;
    for i = 1 to N do
        Calculate the p value by Eq. (20);
        Calculate the F_i^(k) and Cr_i^(k) by Eq. (11)-(13),
          (18) and (19);
        Calculate the rank-based selective pressure rand
          indices by Eq. (9) and (10);
        Generate the mutation trial vectors v_{i,k} by
          Eq. (2) and (5);
        Generate the candidate solution u_{i,j} uses
          Eq. (22);
        Constrain the boundary of individual u_i;
        Evaluate and record the fitness value of the u_i;
        nfes = nfes + 1;
        Select the off-spring individual x_i by Eq. (4);
    end
    Update external archive;
    Update H_F and H_Cr by Eq. (14)-(17);
    Update sub-population racios γ_1^(k+1) and γ_2^(k+1) by
      Eq. (6)-(8);
    Update the population size N^(k+1) by Eq. (21);
end
```

4) Selection:

$$x_i^{(k+1)} = \begin{cases} u_i^{(k)}, & f(u_i^{(k)}) \le f(x_i^{(k)}) \\ x_i^{(k)}, & \text{otherwise} \end{cases} \quad (4)$$

5) External Archive: In the mutation strategy of the basic DE framework we adopt, a random index $r2$ is used to select useful individuals from a joint population consisting of the current population and an external archive. The size of this archive is defined as $Ar \cdot N$, where N represents the number of individuals in the population.

6) DE/current-to-order-pbest/1 Mutation Strategy:

$$\begin{aligned} v_{i,j}^{(k+1)} &= x_{i,j}^{(k)} + F\left(x_{ord\_pbest,j}^{(k)} - x_{i,j}^{(k)}\right) \\ &+ F\left(x_{ord\_median,j}^{(k)} - x_{ord\_worst,j}^{(k)}\right) \end{aligned} \quad (5)$$

7) Hybridization of Two Mutation Strategies:

$$\omega_{m1}^{(k)} = \frac{\sum_{i=1}^{N_1}(f_{m1,i}^{(k)} - f_{m1,i}^{(k-1)})}{N_1} \quad (6)$$

$$\omega_{m2}^{(k)} = \frac{\sum_{i=1}^{N_2}(f_{m2,i}^{(k)} - f_{m2,i}^{(k-1)})}{N_2} \quad (7)$$

$$\gamma_1^{(k+1)} = \begin{cases} 0.5, & \omega_{m1}^{(k)} = \omega_{m2}^{(k-1)} = 0 \text{ or } k = 1 \\ \frac{\omega_{m1}^{(k)}}{\omega_{m1}^{(k)} + \omega_{m2}^{(k-1)}}, & \text{otherwise} \end{cases} \quad (8)$$

8) Extened Rank-based Selective Pressure Strategy:

$$Rank_i = k_r \cdot (N - i) + 1 \quad (9)$$

$$pr_i = Rank_i / (Rank_1 + Rank_2 + ... + Rank_N) \quad (10)$$

9) Success Historical Memory based Parameter Adaptive Strategy:

$$Cr_i^{(k)} = CauchyRand(\mu_{F,h}, 0.1) \quad (11)$$

$$F_i^{(k)} = NormalRand(\mu_{Cr,h}, 0.1) \quad (12)$$

$$h = mod(\frac{k}{H}) \quad (13)$$

$$\mu_{F,j}^{(k)} = \frac{\sum_{n=1}^{|S_F^{(k)}|} \omega_n^{(k)} {s_{F,n}^{(k)}}^2}{\sum_{n=1}^{|S_F^{(k)}|} \omega_n^{(k)} s_{F,n}^{(k)}} \quad (14)$$

$$\mu_{Cr,j}^{(k)} = \frac{\sum_{n=1}^{|S_{Cr}^{(k)}|} \omega_n^{(k)} {s_{Cr,n}^{(k)}}^2}{\sum_{n=1}^{|S_{Cr}^{(k)}|} \omega_n^{(k)} s_{Cr,n}^{(k)}} \quad (15)$$

$$\omega_n^{(k)} = \frac{f_n^{(k)} - f_n^{(k-1)}}{\sum_{g=1}^{|S^{(k)}|} f_g^{(k)} - f_g^{(k-1)}} \quad (16)$$

$$H_{F,H} = H_{Cr,H} = 0.9 \quad (17)$$

$$F_i^{(k)} = 0.7, \quad \text{if } nfes < 0.6 \cdot max\_nfes \text{ and } F_i^{(k)} > 0.7 \quad (18)$$

$$Cr_i^{(k)} = \begin{cases} 0.7, & \text{if } nfes < 0.25 \cdot max\_nfes \\ & \text{and } F_i^{(k)} < 0.7 \\ 0.6, & \text{else if } nfes < 0.5 \cdot max\_nfes \\ & \text{and } F_i^{(k)} < 0.6 \end{cases} \quad (19)$$

10) Linear p Value Reduction:

$$p^{(k)} = p_{max} \cdot (1 - 0.5 \cdot \frac{nfes}{max\_nfes}) \quad (20)$$

11) Linear Population Size Reduction:

$$N^{(k+1)} = round\,((\frac{N_{min}^{(k)} - N_{max}^{(k)}}{max\_nfes}) \cdot nfes + N_{max}^{(k)}) \quad (21)$$

12) Cauthy Perturbation:

$$u_{i,j}^{(k)} = \begin{cases} v_{i,j}^{(k)}, & j = j_{rand} \text{ or } rand < Cr_i \\ CauchyRand(x_{i,j}^{(k)}, 0.1), & \text{else if } rand < p_r \\ x_{i,j}^{(k)}, & \text{otherwise} \end{cases} \quad (22)$$

II. Experimental Results

A. Benchmark Functions

The performance of the proposed algorithm RDE is tested on the CEC 2024 Competition for Single-Objective Real Parameter Numerical Optimization [2]. The benchmark suite, CEC2024, comprises 29 test functions (with $30D$) with diverse features, including unimodal functions ($f_1 - f_2$), multimodal functions ($f3 - f9$), hybrid funtions ($f10 - f19$), and composition functions ($f20 - f29$). The variables in hybrid functions are randomly divided into subcomponents, and different basic functions are used for each subcomponent. The composition functions utilize basic benchmark functions to create problems with a randomly located global optimum and several randomly located deep local optima, making them more challenging to solve.

B. Parameter Settings

The algorithm parameter settings are set as follows:

1) population maximum and minimum sizes $N_{min} = 4$ and $N_{max} = 18 \cdot D$;
2) success history archive memory size $H = 5$;
3) rank greediness factor $k_r = 3$;
4) external archive size $Ar = 1$;
5) differential mutation p value: $p_{max} = 0.25$;
6) initialized sub-population resource ratios $\gamma_1 = \gamma_2 = 0.5$;
7) initialized scale factor and crossover rate in memory archive $\mu_F = 0.3$ and $\mu_{Cr} = 0.8$;
8) Cauthy pertubation strategy rate $p_r = 0.2$;

C. Experimental Settings

According to the guidelines of CEC2024, 25 independent runs are implemented. The maximum evaluation number is set to $max\_nfes = 10000 \cdot D$, where the D is the dimensionality of the optimization problems. The search range is set to $[-100, 100]^D$. Fitness errors less than $1 \times 10^{`8}$ are considered to 0. The experiments were performed using the following

TABLE I
EXPERIMENTAL COMPARISON RESULTS BETWEEN RDE AND OTHER COMPETITORS ON THE 29 BENCHMARK FUNCTIONS IN CEC2024.

Problem	RDE		LSHADE-RSP			iLSHADE-RSP			HSES			EBOwithCMAR			LSHADE		
	Mean	SD	Mean	SD	W	Mean	SD	W	Mean	SD	W	Mean	SD	W	Mean	SD	W
1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**
2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**
3	2.16E+01	1.53E+00	5.86E+01	0.00E+00	+	2.23E+01	8.92E-01	+	**2.66E+00**	**1.90E+00**	-	5.65E+01	1.11E+01	+	5.86E+01	3.41E-14	+
4	7.65E+00	1.85E+00	8.43E+00	3.48E+00	=	6.89E+00	1.62E+00	-	8.84E+00	3.76E+00	=	**2.78E+00**	**1.74E+00**	-	6.70E+00	1.40E+00	-
5	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**=**	1.34E-08	4.11E-08	+	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	2.68E-09	1.92E-08	=
6	3.86E+01	1.73E+00	4.12E+01	3.83E+00	+	3.89E+01	1.69E+00	=	3.91E+01	3.20E+00	=	**3.35E+01**	**8.37E-01**	**-**	3.73E+01	1.37E+00	-
7	8.37E+00	1.92E+00	9.29E+00	4.36E+00	=	7.74E+00	1.77E+00	=	7.65E+00	2.74E+00	-	**2.02E+00**	**1.32E+00**	**-**	6.94E+00	1.75E+00	-
8	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**	**0.00E+00**	**0.00E+00**	**=**
9	1.43E+03	2.43E+02	2.87E+03	8.64E+02	+	1.68E+03	2.70E+02	+	**9.63E+02**	**3.37E+02**	**-**	1.41E+03	2.15E+02	=	1.41E+03	2.35E+02	=
10	3.93E+00	3.38E+00	**3.13E+00**	**8.50E+00**	**-**	3.30E+00	5.56E+00	=	1.79E+01	2.59E+01	+	4.49E+00	8.77E+00	=	3.20E+01	2.89E+01	+
11	2.50E+02	1.66E+02	9.66E+01	7.48E+01	-	1.19E+02	8.37E+01	-	**3.84E+01**	**1.05E+02**	**-**	4.63E+02	2.63E+02	+	1.13E+03	3.17E+02	+
12	**1.44E+01**	**5.80E+00**	1.71E+01	3.61E+00	+	1.81E+01	4.54E+00	+	2.77E+01	1.29E+01	+	1.49E+01	6.25E+00	=	1.51E+01	5.46E+00	=
13	2.02E+01	5.98E+00	2.15E+01	1.15E+00	=	2.16E+01	1.14E+00	=	**1.35E+01**	**9.87E+00**	**-**	2.19E+01	3.84E+00	+	2.07E+01	4.89E+00	=
14	1.46E+00	8.86E-01	**8.25E-01**	**5.36E-01**	-	1.05E+00	6.39E-01	=	4.72E+00	4.74E+00	+	3.69E+00	2.15E+00	+	3.05E+00	1.39E+00	+
15	2.72E+01	4.49E+01	**1.44E+01**	**5.46E+00**	**=**	1.65E+01	5.25E+00	+	2.51E+02	2.16E+02	+	4.26E+01	5.69E+01	+	4.87E+01	4.35E+01	+
16	2.94E+01	1.10E+01	3.80E+01	9.81E+00	+	3.55E+01	5.04E+00	+	**2.46E+01**	**3.53E+01**	**-**	2.98E+01	7.50E+00	=	3.32E+01	5.36E+00	=
17	2.03E+01	2.86E+00	2.08E+01	2.32E-01	+	2.00E+01	3.88E+00	+	**1.96E+01**	**4.96E+00**	**+**	2.21E+01	1.09E+00	+	2.19E+01	1.15E+00	+
18	3.35E+00	9.36E-01	**3.21E+00**	**7.47E-01**	=	3.34E+00	7.47E-01	=	4.07E+00	2.00E+00	=	8.04E+00	2.28E+00	+	4.92E+00	1.61E+00	+
19	**2.57E+01**	**6.66E+00**	2.65E+01	7.16E+00	=	3.22E+01	5.81E+00	+	1.44E+02	3.17E+01	+	3.57E+01	7.50E+00	+	3.20E+01	5.97E+00	+
20	2.08E+02	2.05E+00	2.09E+02	4.07E+00	+	2.08E+02	1.97E+00	=	2.08E+02	3.74E+00	=	**1.99E+02**	**2.02E+01**	**-**	2.07E+02	1.30E+00	=
21	**1.00E+02**	**0.00E+00**	**1.00E+02**	**0.00E+00**	**=**	**1.00E+02**	**0.00E+00**	**=**	**1.00E+02**	**0.00E+00**	**=**	**1.00E+02**	**0.00E+00**	**=**	1.00E+02	1.00E-13	+
22	**3.46E+02**	**3.38E+00**	3.52E+02	4.14E+00	+	3.51E+02	3.04E+00	+	3.52E+02	8.59E+00	+	3.51E+02	3.51E+00	+	3.50E+02	2.92E+00	+
23	4.23E+02	2.38E+00	4.27E+02	2.68E+00	+	4.26E+02	1.87E+00	+	4.21E+02	3.39E+00	-	**4.18E+02**	**4.55E+01**	**+**	4.26E+02	1.70E+00	+
24	**3.79E+02**	**1.56E-01**	3.87E+02	8.02E-03	+	3.79E+02	2.29E-01	+	3.87E+02	2.65E-02	+	3.87E+02	7.56E-01	+	3.87E+02	2.28E-02	+
25	8.94E+02	4.27E+01	9.01E+02	4.03E+01	=	9.24E+02	3.89E+01	+	8.77E+02	2.00E+02	=	**5.37E+02**	**3.06E+02**	**-**	9.17E+02	3.30E+01	+
26	**4.73E+02**	**6.62E+00**	4.96E+02	8.00E+00	+	4.77E+02	6.21E+00	+	5.20E+02	9.26E+00	+	5.02E+02	4.03E+00	+	5.03E+02	6.46E+00	+
27	3.18E+02	4.14E+01	**3.02E+02**	**1.60E+01**	**-**	3.04E+02	2.23E+01	-	3.18E+02	3.98E+01	=	3.08E+02	2.88E+01	=	3.22E+02	4.63E+01	+
28	**3.88E+02**	**2.66E+01**	4.38E+02	1.84E+01	+	3.92E+02	2.08E+01	+	4.55E+02	5.40E+01	+	4.33E+02	1.13E+01	+	4.32E+02	6.13E+00	+
29	**8.48E+02**	**3.99E+02**	1.97E+03	1.04E+01	+	1.01E+03	3.35E+02	+	2.06E+03	5.12E+01	+	1.99E+03	4.21E+01	+	1.98E+03	3.79E+01	+
W/T/L	−/−/−		13/13/4			15/12/3			11/12/7			14/11/5			17/10/3		

system: CPU: Intel® Core™i7-12700kf 3.6GHz, 32GB DDR4-3600MHz RAM, Programming Language: C++, and Operator System: Windows 10.

D. Statistical Results

The Wilcoxon rank-sum test is utilized to determine if there is a statistically significant difference between two algorithms in solving a problem, with a significance level of $\alpha = 0.05$. Each problem is tested 51 times independently in a row. The symbol "$+$" denotes the superior algorithm when there is a significant difference between the pair, while "$-$" indicates the relatively inferior one. "$=$" signifies that no significant difference exists between the pair of algorithms.

The performance of the RDE was compared to the other compatitors which showed excellent performance on previous CEC benchmark suites, including:

1) LSHADE-RSP [3]: LSHADE Algorithm with Rank-Based Selective Pressure Strategy for Solving CEC 2017 Benchmark Problems;
2) iLSHADE-RSP [4]: An improved LSHADE-RSP algorithm with the Cauchy perturbation;
3) HSES [5]: Hybrid Sampling Evolution Strategy for Solving Single Objective Bound Constrained Problems;
4) EBOwithCMAR [6]: Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat phase;
5) LSHADE [7]: Improving the search performance of SHADE using linear population size reduction.

As shown in Table I, RDE demonstrates superiority over competitive algorithms on CEC2024. In addition, RDE also performs well on relatively more complex composition functions ($f20 - f29$), indicating the effectiveness of recombination research. Compared with LSHADE-RSP, iLSHADE-RSP, HSES, EBOwithCMAR and LSHADE, RDE achieves better solutions on 13, 15, 11, 14 and 17 problems respectively and fails to beat opponents on 4, 3, 7, 5 and 3 problems respectively. Although RDE shows significant advantages over advanced variants of the DE series, the performance gap is not significantly widened compared to HSES.

REFERENCES

[1] S. Tao, R. Zhao, K. Wang, and S. Gao, "An efficient reconstructed differential evolution variant by some of the current state-of-the-art strategies for solving single objective bound constrained problems," *arXiv preprint arXiv:2404.16280*, 2024.

[2] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization," *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*, 2017.

[3] V. Stanovov, S. Akhmedova, and E. Semenkin, "LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[4] T. J. Choi and C. W. Ahn, "An improved lshade-rsp algorithm with the cauchy perturbation: ilshade-rsp," *Knowledge-Based Systems*, vol. 215, p. 106628, 2021.

[5] G. Zhang and Y. Shi, "Hybrid sampling evolution strategy for solving single objective bound constrained problems," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–7.

[6] A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 1835–1842.

[7] 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.