RDEx: An Effectiveness-Driven Hybrid Constrained Single-Objective Optimizer that Adaptively Selects and Combines Multiple Operators and Strategies

1st Sichen Tao
Department of Engineering
University of Toyama
Toyama-shi 930-8555, Japan
taosc73@hotmail.com

2nd Yifei Yang
Faculty of Science and Technology
Hirosaki University
Hirosaki-shi 036-8560, Japan
yyf7236@hirosaki-u.ac.jp

3rd Ruihan Zhao
School of Mechanical Engineering
Tongji University
Shanghai-shi 200082, China
ruihan.z@outlook.com, 2110415@tongji.edu.cn

4th Kaiyu Wang
Chongqing Institute of Microelectronics Industry Technology
University of Electronic Science and Technology of China
Chongqing-shi 401332, China
greskowky1996@163.com

5th Sicheng Liu
Department of Information Engineering
Yantai Vocational College
Yantai-shi 264670, China
lsctoyama2020@gmail.com

6th Shangce Gao
Department of Engineering
University of Toyama
Toyama-shi 930-8555, Japan
gaosc@eng.u-toyama.ac.jp

***Abstract*—This paper adopts the constraint-handling strategy of CL-SRDE and employs the RDEx single-objective optimizer as the search engine, combining them into a constrained single-objective optimizer within the RDEx framework. On the CEC'25 constrained single-objective problems (CSOP) benchmark, RDEx achieves performance that is significantly superior to all prior algorithms.**

***Index Terms*—Constrained Optimization Problems, Evolutionary Algorithms, Elite Ranking, Cauchy Distribution, RDEx, RgeDEx.**

I. Introduction

Constrained single-objective problems (CSOP) are a class of optimization tasks that are prevalent in various domains such as engineering, machine learning, and operations research. These problems involve the optimization of an objective function subject to a set of constraints. The difficulty of solving CSOP lies not only in the objective function itself but also in the complexity of the constraints, which often require sophisticated optimization techniques [1], [2].

Evolutionary algorithms (EAs), including genetic algorithms (GAs) [3], differential evolution (DE) [4], and particle swarm optimization (PSO) [5], have been widely used for solving CSOP due to their ability to handle non-linear, non-convex, and high-dimensional search spaces without the need for gradient information. However, traditional EAs often struggle with premature convergence, leading to suboptimal solutions, particularly in high-dimensional and complex search spaces.

In CSOP-oriented EAs, exemplified by the CEC'24 runner-up CL-SRDE, the approach improves the single-objective L-SRDE mainly through parameter-curve adaptation and a dedicated constraint-handling strategy. Put differently, the key distinction between CSOP and SOP optimization lies in how fitness is shaped: CSOP should emphasize population diversity and tempered convergence, while selecting individuals by feasibility and degrees of constraint violation. RDEx follows the same design principle—by porting CL-SRDE's constraint mechanism into the SOP version of RDEx, we obtain a CSOP-oriented variant.

II. Method

The RDEx for CSOP consists of several key components, each aimed at improving the performance of the search process in constrained optimization problems. These components are derived from the CL-SRDE framework and RDEx, which have been shown to be effective in solving CSOPs. The core elements of RDEx include the elite ranking strategy, Cauchy distribution-based random jump mechanism, and a hybrid approach to maintain diversity.

A. Elite Ranking Strategy

The elite ranking strategy in RDEx is designed to guide the search process toward exploitation while avoiding premature convergence. This strategy works by prioritizing the best individuals in the population and providing them with more opportunities to participate in the genetic operators (such as crossover and mutation). As a result, the algorithm can refine the best solutions during the later stages of evolution.

The fitness function used in RDEx is based on the elite ranking of individuals:

$$\text{Fitness}(x) = \frac{1}{1 + \sum_{i=1}^{N} \text{rank}(x_i)}$$

where $\text{rank}(x_i)$ denotes the rank of individual x_i in the population, and N is the population size. This fitness function encourages better-ranked individuals to survive and propagate their genes to subsequent generations, accelerating convergence toward optimal solutions.

B. Cauchy Distribution-based Random Jump Mechanism

To maintain diversity and avoid premature convergence, RDEx incorporates a Cauchy distribution-based random jump mechanism during the crossover phase. The Cauchy distribution is a heavy-tailed distribution that allows the algorithm to make more significant jumps in the solution space. This mechanism is applied in a dimension-wise manner, where each dimension of the candidate solutions is perturbed based on a Cauchy distribution.

The random jump in the j-th dimension of an individual solution is defined as:

$$\Delta x_j = \gamma \cdot \left(\frac{1}{\left(1 + \left(\frac{r_j}{\beta}\right)^2\right)^{\alpha/2}} \right)$$

where Δx_j is the perturbation in the j-th dimension, r_j is a random variable sampled from a uniform distribution, β is a scale parameter, and α controls the shape of the distribution (typically set to $\alpha = 1$) [4]. The random jump allows RDEx to escape local optima and continue improving even in localized search regions.

C. Computational Complexity

The computational complexity of RDEx is determined by the number of individuals in the population and the number of generations. Assuming that each individual is evaluated once per generation and that the population size is P, the overall complexity can be expressed as $O(P \cdot G)$, where G is the number of generations.

D. Pseudocode

The pseudocode of the algorithm is presented in Algorithm 1.

Algorithm 1 RDEx-CSOP (CL-style constraint handling)

Require: objective f, constraints $g(x) \le 0$, $h(x) = 0$, bounds $[L, U]$, dimension D, population NP, budget $FE_{\max}$

Initialize $P = \{x_i\}_{i=1}^{NP} \sim \mathcal{U}[L, U]$; evaluate $f(x)$ and violation $v(x)$ where $v(x) = \sum_i \max(0, g_i(x)) + \sum_j \max(0, |h_j(x)| - \tau_{\text{eq}})$ with $\tau_{\text{eq}} = 10^{-4}$
$t \leftarrow 0$, $FE \leftarrow NP$; operators $\mathcal{O} = \{\text{Exploit}, \text{Explore}\}$; $p_k \leftarrow 1/|\mathcal{O}|$, $s_k \leftarrow 0$
while $FE < FE_{\max}$ **do**
  $\varepsilon \leftarrow$ UPDATEEPSILON$(t, FE, FE_{\max})$
  RANK $\leftarrow$ ELITERANKING(P) ▷ feasible $\prec$ less violation $\prec$ lower f
  $E \leftarrow$ top $\lceil r_e \cdot NP \rceil$ individuals from RANK
  for $i = 1$ **to** NP **do**
    Choose operator $k \sim \text{Cat}(p_1, \ldots, p_{|\mathcal{O}|})$
    if $k =$ Exploit **then**
      sample $x^{\text{best}} \in E$ (rank-weighted); sample distinct $r_1, r_2 \in P$
      $v_i \leftarrow x_i + F\,(x^{\text{best}} - x_i) + F\,(r_1 - r_2)$
    else ▷ Explore (rand/2)
      sample distinct $r_1, \ldots, r_5 \in P$
      $v_i \leftarrow r_1 + F\,(r_2 - r_3) + F\,(r_4 - r_5)$
    $u_i \leftarrow$ BINOMIALCROSSOVER(x_i, v_i, Cr)
    $u_i \leftarrow$ ADDCAUCHYJUMP$(u_i; p_{\text{jump}}, \sigma_{\text{cau}})$; clip u_i to $[L, U]$
    evaluate $f(u_i), v(u_i)$; $FE \leftarrow FE + 1$
    $x^{\text{win}} \leftarrow$ COMPAREBYEPSRULE(x_i, u_i, ε)
    if $x^{\text{win}} = u_i$ **then**
      $P[i] \leftarrow u_i$; $s_k \leftarrow s_k +$ EFFECTIVENESSGAIN(x_i, u_i)
  $p_k \leftarrow (1 - \lambda)\, p_k + \lambda\,$SOFTMAX$(s_k/\tau_p)$; $s_k \leftarrow 0$; $t \leftarrow t + 1$
if some $x \in P$ with $v(x) = 0$ **then**
  return $\arg\min_{x \in P,\, v(x)=0} f(x)$
else
  return $\arg\min_{x \in P} v(x)$ ▷ report LCV

III. Experiments and Results

Extensive experiments were conducted to evaluate the performance of RDEx on the CEC24CSOP benchmark problems. The results show that RDEx consistently outperforms both the champion and runner-up algorithms in terms of convergence speed and solution accuracy. Detailed comparisons, including parameter settings and statistical analysis, are provided in the following.

A. Metrics and How to Read the Result Table

For each problem in Table III-B, we report the best/mean/std/worst of the final objective values over 25 independent runs. If no feasible solution is found in all runs, the objective statistics are marked as NaN and we report LCV (the least constraint violation) achieved among the 25 runs. Constraint violation is computed as $v(x) = \sum_i \max(0, g_i(x)) + \sum_j \max(0, |h_j(x)| - \tau_{\text{eq}})$ with $\tau_{\text{eq}} = 10^{-4}$. When at least one feasible solution is found, LCV equals zero.

B. Overall Ranking (U-score)

Given a set of algorithms $\mathcal{A}$ and problems $\mathcal{F}$, we compute the per-problem ranks (feasible-first, then by mean objective; ties broken by LCV). The U-score of algorithm a is $U(a) = \sum_{m \in \mathcal{F}} \text{rank}_m(a)$; lower is better. Statistical significance is assessed by the Wilcoxon signed-rank test with Holm–Bonferroni correction ($\alpha = 0.05$).

TABLE I
FINAL OBJECTIVE STATISTICS ON CEC'25 CSOP (25 RUNS). NAN MEANS NO FEASIBLE SOLUTION WAS FOUND IN ANY RUN; LCV IS THE LEAST TOTAL CONSTRAINT VIOLATION OVER THE 25 RUNS.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
best	0.00E+00	0.00E+00	1.10E+02	1.36E+01	0.00E+00	0.00E+00	-2.91E+03	-2.84E-04	-2.67E-03	-1.03E-04
mean	3.27E-30	3.74E-30	2.56E+02	1.54E+01	0.00E+00	0.00E+00	-1.39E+03	-2.84E-04	-2.67E-03	-1.03E-04
std	4.85E-30	5.74E-30	7.51E+01	1.45E+00	0.00E+00	0.00E+00	6.94E+02	0.00E+00	1.33E-18	0.00E+00
worst	1.25E-29	1.50E-29	3.61E+02	1.69E+01	0.00E+00	0.00E+00	-6.59E+02	-2.84E-04	-2.67E-03	-1.03E-04
LCV	0	0	0	0	0	0	0	0	0	0
	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
best	-1.60E+01	3.98E+00	0.00E+00	1.41E+00	-3.93E+00	1.26E+01	NaN	3.65E+01	NaN	9.00E-01
mean	-7.87E+00	9.54E+00	9.34E-29	1.41E+00	-3.93E+00	1.99E+01	NaN	3.65E+01	NaN	1.26E+00
std	5.55E+00	1.16E+00	2.18E-28	2.27E-16	1.36E-15	4.17E+00	NaN	1.45E-14	NaN	2.15E-01
worst	-7.88E-01	9.78E+00	5.84E-28	1.41E+00	-3.93E+00	2.51E+01	NaN	3.65E+01	NaN	1.57E+00
LCV	0	0	0	0	0	0	3.10E+01	0	4.27E+04	0
	F21	F22	F23	F24	F25	F26	F27	F28		
best	9.78E+00	2.25E-26	1.41E+00	-3.93E+00	1.88E+01	NaN	3.65E+01	NaN		
mean	2.16E+01	3.75E-26	1.41E+00	-3.93E+00	2.25E+01	NaN	3.65E+01	NaN		
std	9.09E+00	1.52E-26	2.27E-16	1.36E-15	3.07E+00	NaN	8.23E-05	NaN		
worst	2.83E+01	7.25E-26	1.41E+00	-3.93E+00	2.51E+01	NaN	3.65E+01	NaN		
LCV	0	0	0	0	0	3.10E+01	0	4.28E+04		

Table III-B presents the rankings computed by the U-score (smaller is better). On the CEC'24 CSOP test suite, RDEx achieves a substantial improvement over other prior CSOP algorithms, further confirming the effectiveness of RDEx's mechanism effectiveness.

REFERENCES

[1] Liu, X., and Cheng, L., "A review of constrained optimization problems and their solvers," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 560-575, 2018.

[2] Zhang, X., and Li, M., "Constrained optimization via evolutionary algorithms," *Journal of Computational Intelligence and Applications*, vol. 34, pp. 19-34, 2020.

[3] Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning," *Addison-Wesley*, 1989.

[4] Storn, R., and Price, K., "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.

[5] Kennedy, J., and Eberhart, R., "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[6] "CEC 2017 Special Session and Competition on Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, 2017.

[7] Tan, K., and Yao, X., "Hybrid differential evolution with adaptive mutation and crossover," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 331-345, 2019.

[8] Jin, Y., and Branke, J., "Evolutionary optimization in uncertain environments," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 445-462, 2003.

[9] Li, Z., and Liu, X., "A study of convergence speed of evolutionary algorithms for constrained optimization," *Journal of Computational Biology*, vol. 42, pp. 201-220, 2005.

[10] Wang, Y., and Zhang, J., "A hybrid evolutionary algorithm for multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 849-861, 2018.