Closed-Loop Success-History Driven Reconstructed Differential Evolution for Constrained Multi-Objective Optimization

1st Chennuo Jin
Faculty of Science and Technology
Hirosaki University
Hirosaki-shi, 036-8560, Japan
hn25rs01@hirosaki-u.ac.jp

2nd Bingchao Shi
Faculty of Science and Technology
Hirosaki University
Hirosaki-shi, 036-8560, Japan
h25ms415@hirosaki-u.ac.jp

3rd Sichen Tao
Cyberscience Center
Tohoku University
Sendai-shi, 980-8578, Japan
sichen.tao@tohoku.ac.jp

4th Yifei Yang*
Faculty of Science and Technology
Hirosaki University
Hirosaki-shi, 036-8560, Japan
yyf7236@hirosaki-u.ac.jp

***Abstract*—Constrained multi-objective optimization (CMOP) is challenging because the search must balance feasibility maintenance, convergence toward the constrained Pareto-optimal set, and diversity preservation under complex constraints and limited evaluation budgets. This paper proposes Closed-Loop Success-History Driven Reconstructed Differential Evolution (CLSHRDE), which is built upon the Reconstructed Differential Evolution (RDE) baseline under the Institute of Electrical and Electronics Engineers (IEEE) Congress on Evolutionary Computation (CEC) 2025 CMOP benchmark setting, where RDE has demonstrated strong overall performance. On this basis, CLSHRDE introduces an explicit two-operator search structure by integrating an extra stochastic mutation rule with the original order-based mutation. To enable robust stage-wise resource allocation between the two operators, CLSHRDE develops a closed-loop scheduling mechanism driven by an exponentially weighted success-history estimate, providing noise-attenuated feedback for online operator regulation across search stages. Extensive experiments on the Scalable Decision space Constraints (SDC) benchmark suite and real-world constrained multi-objective problems (RWMOPs) show that, under unified parameter settings, CLSHRDE achieves competitive or improved performance in terms of Inverted Generational Distance (IGD), Hypervolume (HV), and feasible success rate (FSR), with improvements statistically supported by one-sided Wilcoxon rank-sum tests.**

***Index Terms*—differential evolution, constrained multi-objective optimization, success-history modeling, closed-loop scheduling, operator competition, feasibility handling.**

I. Introduction

Complex optimization problems frequently arise in engineering design, control systems, hyperparameter tuning, and industrial decision-making, and are often non-differentiable, non-convex, multi-modal, constrained, and expensive to evaluate [1]. Compared to gradient-based or model-based approaches, evolutionary algorithms (EAs) do not require gradients, explicit models, or analytical expressions, making them well suited for practical application [2].

Constrained multi-objective optimization (CMOP) is an important class of problems in evolutionary computation, aiming to approximate a high-quality Pareto front under explicit feasibility requirements [3]. Compared to constrained single-objective optimization (CSOP), CMOPs are generally more challenging because the search must simultaneously satisfy three coupled demands: feasibility maintenance, convergence toward the Pareto-optimal set, and preservation of diversity along the front [4]. This difficulty is further amplified by two factors. First, as dimensionality increases, the search space expands rapidly, often growing exponentially with dimension [5], while the proportion of mutually non-dominated solutions typically increases, which weakens selection pressure and makes it harder to drive the population toward the Pareto-optimal set [6]. Second, explicit constraints often shrink and fragment the feasible region, rendering feasible-solution discovery and long-term feasibility maintenance substantially more difficult [7]. Consequently, CMOP solvers typically exhibit stage-dependent behaviors, including early feasibility exploration, subsequent Pareto-front expansion [8], and late-stage convergence refinement [9], under a more delicate feasibility–convergence-diversity tradeoff than in CSOP.

Such characteristics necessitate CMOP solvers that can adaptively allocate limited evaluations to balance feasibility maintenance, convergence, and diversity across evolutionary stages. Differential evolution (DE) is well suited to serve as a baseline framework in this context due to its structural simplicity and search efficiency, and it has been widely used in constrained multi-objective optimization [10]. Based on this baseline, numerous DE-based CMOP variants have been proposed by improving mutation and crossover designs, parameter adaptation, and population management mechanisms [9]. To further exploit complementary search behaviors, some studies introduce multi-operator or multi-branch structures with operator-selection and scheduling mechanisms [9]. On the Institute of Electrical and Electronics Engineers (IEEE) Congress on Evolutionary Computation (CEC) 2025 CMOP competition benchmarks, Reconstructed Differential Evolution (RDE) achieves leading overall performance [11], indicating strong search capability in constrained environments. Nevertheless, RDE does not explicitly incorporate a systematic multi-operator competition and online regulation mechanism, which may limit its adaptability to the stage-dependent demands of CMOP solving.

Motivated by this, we propose Closed-Loop Success-History Driven Reconstructed Differential Evolution (CLSHRDE), which adopts the RDE baseline used in the CEC constrained multi-objective benchmark setting [11]. Here, the RDE base-

*Corresponding author: Yifei Yang (yyf7236@hirosaki-u.ac.jp).

line refers to the implementation/configuration adopted in the CEC CMOP setting and is different from the original single-objective RDE proposed for bound-constrained optimization [12]. Building upon this baseline, CLSHRDE introduces two complementary search operators as distinct search branches, and further incorporates a closed-loop success-history-driven competition framework to dynamically regulate exploration and exploitation throughout the evolutionary process. Consequently, CLSHRDE is designed to better support feasibility acquisition, front expansion, and convergence refinement across different evolutionary stages. Experiments on the CEC multi-objective constrained benchmark suite [13] and classical real-world constrained multi-objective engineering problems (RWMOPs) [14] show that, under unified parameter settings, CLSHRDE outperforms several state-of-the-art algorithms in terms of Inverted Generational Distance (IGD), Hypervolume (HV) [15], and the observed improvements are statistically supported by Wilcoxon rank-sum tests [16].

The main contributions of this work are summarized as follows:

- We propose Closed-Loop Success-History Driven Reconstructed Differential Evolution (CLSHRDE), an extension of RDE that combines the original order-based mutation with an additional stochastic mutation to form a simple two-operator search structure.
- We introduce operator transplantation and an exponentially weighted success-history estimator, and develop a time-dependent success-rate feedback mechanism to regulate operator usage online.
- We formulate a Closed-Loop Success-Driven Competition Across Search Stages and validate CLSHRDE on the Scalable Decision space Constraints (SDC) benchmark suite and RWMOPs using IGD, HV, and feasible success rate (FSR), with pairwise comparisons conducted by one-sided Wilcoxon rank-sum tests.

The remainder of this paper is organized as follows. Section II reviews differential evolution and related work. Section III introduces the CLSHRDE algorithm. Section IV presents the experimental setup and results. Section V concludes the paper.

II. RELATED WORK

In this section, we briefly introduce DE [17], its advanced variant Linear Population Size Reduction Success-History-Based Adaptive Differential Evolution (LSHADE) [18], and Enhanced Binary Learning Success-History-Based Adaptive Differential Evolution (EBLSHADE) [19]. The subsequent sections progressively develop the proposed method based on this basic DE framework. DE generates new candidate solutions through mutation and crossover [17]. A representative mutation strategy is the random-based differential mutation:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), \tag{1}$$

where r_1, r_2 and r_3 are distinct indices and F denotes the scaling factor.

LSHADE has been recognized as one of the most effective DE algorithms. LSHADE combines success-history-Based parameter adaptation with a linear population size reduction strategy, which gradually decreases the population size during evolution. This mechanism preserves population diversity in the early stages while strengthening exploitation as the search progresses. In LSHADE, a commonly adopted mutation scheme can be expressed as

$$\mathbf{v}_i^{(k+1)} = \mathbf{x}_i^{(k)} + F\big(\mathbf{x}_{\text{pbest}}^{(k)} - \mathbf{x}_i^{(k)}\big) + F\big(\mathbf{x}_{r_1}^{(k)} - \mathbf{x}_{r_2}^{(k)}\big), \tag{2}$$

where k denotes the iteration index, $\mathbf{x}_{\text{pbest}}^{(k)}$ is randomly selected from the top-performing individuals in the current population, and $\mathbf{x}_{r_1}^{(k)}$ and $\mathbf{x}_{r_2}^{(k)}$ are two mutually distinct individuals randomly selected from the population.

EBLSHADE instead revises the internal composition of differential vectors to enhance search stability. One representative component of EBLSHADE is the incorporation of the DE/current-to-order-pbest/1 mutation strategy, expressed as

$$\begin{aligned} v_{i,j}^{(k+1)} = x_{i,j}^{(k)} &+ F\big(x_{\text{ordpbest},j}^{(k)} - x_{i,j}^{(k)}\big) \\ &+ F\big(x_{\text{ordmedian},j}^{(k)} - x_{\text{ordworst},j}^{(k)}\big). \end{aligned} \tag{3}$$

where $x_{\text{ordpbest}}^{(k)}$, $x_{\text{ordmedian}}^{(k)}$, and $x_{\text{ordworst}}^{(k)}$ denote the best, median, and worst individuals after fitness-based ordering. By integrating these previously proposed operators and mechanisms, RDE achieves superior performance compared with these existing algorithms.

III. PROPOSED APPROACH: CLSHRDE

A. Operator Transplantation and Exponentially-Weighted Success History

a) Operator Transplantation: To introduce complementary search behaviors over the RDE baseline, we incorporate an extra stochastic mutation rule in the trial-vector generation stage, while keeping the underlying RDE mutation scheme unchanged. The incorporated rule is defined as

$$\mathbf{v}_i^{(k+1)} = \mathbf{x}_i^{(k)} + F\big(\mathbf{x}_{\text{pbest}}^{(k)} - \mathbf{x}_i^{(k)}\big) + \alpha F\big(\mathbf{x}_{r_1}^{(k)} - \mathbf{x}_{r_2}^{(k)}\big), \tag{4}$$

where $\mathbf{x}_{r_1}^{(k)}$ and $\mathbf{x}_{r_2}^{(k)}$ are randomly selected individuals. The coefficient α regulates the influence of the random differential component, preventing excessive stochastic perturbations while preserving exploratory capability. Together with the order-based mutation in RDE, these two complementary search behaviors provide the structural basis for adaptive resource allocation driven by success-history feedback.

b) Exponentially-Weighted Success History: In classical adaptive DE variants, operator preference is often determined by short-term success statistics, making resource allocation highly sensitive to noise and prone to overemphasizing transient successes while neglecting long-term trends; to obtain more stable and informative feedback, we introduce an exponentially weighted success history for two abstract operators $\mathcal{O}_1$ and $\mathcal{O}_2$ [20], where the exponentially decaying kernel $\lambda(1-\lambda)^{\ell}$ acts as a low-pass filter (inspired by long-memory smoothing in signal processing and fractional dynamics) to suppress short-lived fluctuations and emphasize temporally consistent performance patterns, thereby stabilizing adaptive operator scheduling. This exponentially decaying kernel can be related to memory kernels arising in fractional-order difference models. In particular, while fractional-order systems are typically characterized by power-law memory decay, the geometric decay employed here provides a computationally efficient approximation of long-memory behavior. From this perspective, the proposed success-history mechanism can be viewed as a simplified discrete realization of fractional-order filtering, where recent observations are emphasized while still retaining a diminishing influence from past states. This interpretation offers a theoretical connection between success-history adaptation and long-memory dynamics, and helps explain the improved robustness against noisy short-term fluctuations [21].

Let $r_k^{(u)} \in [0,1]$ denote the per-generation success rate of operator O_k at generation u, where $k \in \{1,2\}$. The proposed success estimate is defined as the exponentially-weighted infinite series

$$s_k^{(u)} = \sum_{\ell=0}^{\infty} \lambda(1-\lambda)^{\ell}\, r_k^{(u-\ell)}, \tag{5}$$

where $\lambda \in (0,1)$ controls the depth of historical information and ℓ denotes the lag index. Due to geometric decay, the formulation admits the recursive update

$$s_k^{(u+1)} = (1-\lambda)s_k^{(u)} + \lambda r_k^{(u)}, \tag{6}$$

which provides an online and computationally efficient update mechanism.

Compared with single-step or short-term success feedback, the exponentially weighted estimate smooths noisy fluctuations and emphasizes longer-term trends, providing a more reliable signal for adaptive operator selection and mitigating premature bias toward operators that succeed only by early chance.

B. Success-Rate Feedback Mechanism and Time-Dependent Scheduling

To regulate the competition between two abstract mutation operators $\mathcal{O}_1$ and $\mathcal{O}_2$, we treat their long-memory success-rate estimates (s_1, s_2) as credit signals for adaptive resource allocation. Unlike instantaneous success counts, these smoothed rates provide noise-attenuated and temporally consistent indicators at the evolutionary time scale. The normalized relative advantage is defined as

$$\Delta s(t) = \frac{s_1 - s_2}{s_1 + s_2 + \epsilon(t)}, \tag{7}$$

where $\Delta s(t) \in [-1,1]$ and $\epsilon(t)$ is a stage-dependent stabilizer used to avoid numerical sensitivity when $s_1 + s_2$ becomes small. We set

$$\epsilon(t) = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min})\exp(-\beta t), \quad t \in [0,1], \tag{8}$$

so that stronger stabilization is applied in early stages and gradually relaxed toward termination.

To reduce premature and over-confident switching in early stages, we introduce an evidence-based confidence weight

$$\rho = \min\left(1, \frac{s_1 + s_2}{\tau}\right), \tag{9}$$

where $\rho \in [0,1]$ increases with the accumulated evidence $(s_1 + s_2)$ and scales the influence of Δ_s until sufficient feedback is collected. The evolutionary stage variable

$$t = \frac{FE}{maxFEs}, \qquad \phi(t) = t^2(3-2t), \tag{10}$$

models the natural transition from exploration to exploitation. Integrating these components, the allocation kernel is defined as

$$\begin{aligned} h(t) &= p_0 + a_T\big(\phi(t) - \tfrac{1}{2}\big) \\ &\quad + \kappa\, \Delta_{\mathrm{s}}\, \rho\, \big(0.1 + 0.9\,\phi(t)^2\big), \end{aligned} \tag{11}$$

and the final selection probabilities are obtained via clipping:

$$P_1(t) = \mathrm{clip}_{[p_{\min}, p_{\max}]}\big(h(t)\big), \qquad P_2(t) = 1 - P_1(t). \tag{12}$$

The factor $(0.1 + 0.9\phi(t)^2) \in [0.1, 1]$ provides a bounded stage-wise gate, ensuring nonzero early feedback and progressively strengthening the advantage term. This yields a closed-loop scheduling mechanism in which credit-driven competition [22], confidence modulation, and stage-awareness jointly regulate the allocation of computational resources.

C. Closed-Loop Success-Driven Competition Across Search Stages

This subsection focuses on the overall impact of the proposed closed-loop success-driven competition mechanism on the evolutionary search process, rather than on any specific mutation operator. By integrating success-rate feedback, confidence modulation, and time-dependent scheduling, the framework enables smooth and adaptive regulation of multiple search branches, leading to stable and stage-aware search dynamics. Moreover, since this closed-loop scheduling philosophy is independent of operator-specific designs, it exhibits strong generality and can be readily extended to a wide range of multi-branch evolutionary frameworks.

From a process-oriented perspective, the resulting evolutionary behavior can be naturally characterized into three successive stages [23].

a) Early stage (exploration-oriented): At the early stage of evolution, the time-dependent phase function $\phi(t)$ suppresses excessive exploitation, preventing premature dominance of any single search branch and preserving behavioral diversity, which is crucial for constrained problems with narrow or disconnected feasible regions.

b) Middle stage (success and confidence-driven): As evolution progresses, the effect of time-dependent suppression weakens, and resource allocation becomes increasingly driven by empirical success rates and confidence signals. Search branches with consistent improvements are preferentially allocated resources, enabling a smooth transition from exploration to exploitation.

Algorithm 1: CLSHRDE

Input: D, N, maxFEs, $F(\cdot)$, $\Phi(\cdot)$
Output: PF^*

Parameters and states:
  Set dimension D and population size N;
  Set success scores $s_1 = 0.5,\ s_2 = 0.5$;
  Set control parameters $p_0, a_T, \kappa, \tau, \lambda, p_{\min}, p_{\max}$;

Initialize:
  Initialize population $P = \{x_i\}_{i=1}^N$;
  Initialize branch indicator vector $\mathbf{b} \in \{0,1\}^N$;
  Evaluate $F(P)$ and $\Phi(P)$;
  Extract feasible set $X_f = \{x \in P \mid \Phi(x) = 0\}$;
  Set $PF \leftarrow \mathrm{ND}(X_f)$;
  Initialize function evaluation counter $FE \leftarrow N$;

Main loop:
while $FE <$ maxFEs **do**
  Phase scheduling:
  Compute $t = FE/\mathrm{maxFEs}$ and $\phi(t) = t^2(3-2t)$;
  Update $\epsilon(t)$ according to the time-dependent rule;
  Credit and allocation:
  Compute $\Delta_s = \dfrac{s_1 - s_2}{s_1 + s_2 + \epsilon(t)}$ and $\rho = \min\left(1, \dfrac{s_1 + s_2}{\tau}\right)$;
  Compute $\mathrm{base} = p_0 + a_T(\phi(t) - 0.5)$;
  Compute $\mathrm{bias} = \kappa\,\Delta_s\,(0.1 + 0.9\,\phi(t)^2)\,\rho$;
  Compute $p = \mathrm{clip}(\mathrm{base} + \mathrm{bias},\ p_{\min}, p_{\max})$;
  Branch competition:
  for $i = 1$ **to** N **do**
    draw $b_i \sim \mathrm{Bernoulli}(p)$;
    if $b_i = 1$ **then**
      $u_i \leftarrow \mathrm{DE/current\text{-}to\text{-}order\text{-}pbest/1}(p_i)$;
    else
      $u_i \leftarrow \mathrm{DE/current\text{-}to\text{-}pbest/1}(\alpha)(p_i)$;
  Success feedback:
    Generate offspring set $U = \{\mathbf{u}_i\}_{i=1}^N$;
    Evaluate $F(U)$ and $\Phi(U)$;
    Compute improvements $\delta_i = \mathbb{I}(u_i \prec p_i)$;
    Compute branch-wise success ratios r_1, r_2;
    Update $s_1 \leftarrow (1-\lambda)s_1 + \lambda r_1$ and $s_2 \leftarrow (1-\lambda)s_2 + \lambda r_2$;
  Environmental selection:
    $P = \mathrm{Select}_{\epsilon(t)}(P \cup \{u_i\}_{i=1}^N)$;
    $FE \leftarrow FE + N$;
end while
return P

c) Late stage (exploitation-oriented): In the late stage, the scheduler favors exploitative branches with stable success histories, and the smoothed feedback avoids abrupt switching to yield steadier convergence for high-precision constrained optimization.

The proposed framework involves a small number of control parameters associated with success-history estimation and scheduling. In this work, these parameters are fixed across all benchmark problems without problem-specific tuning, and the consistent performance across different problems suggests that the method is not overly sensitive to parameter settings.

Overall, embedding multi-branch search behaviors into a closed-loop control structure enables stage-dependent adaptation without handcrafted phase transitions, providing a general mechanism for balancing exploration and exploitation.

IV. Experiments and Results

A. Experimental setup

a) Comparison algorithms: To ensure the timeliness and representativeness of the experimental comparison, this study focuses on state-of-the-art constrained multi-objective optimization algorithms proposed in recent IEEE CEC related studies. These representative methods span diverse algorithmic design philosophies and constraint-handling mechanisms, enabling a multi-perspective evaluation of the proposed approach.

Specifically, the comparison includes RDE, DESDE [24], IMTCMO [13], and A-CCEMT [25], which represent structured differential evolution, multi-population co-evolution, and adaptive competitive constraint-handling paradigms, respectively. It is worth noting that LSHADE and EBLSHADE are primarily developed for single-objective optimization. Since this work focuses on constrained multi-objective optimization, these algorithms are not directly applicable without significant modifications. Therefore, we follow the CEC CMOP benchmarking practice and compare with state-of-the-art CMOP algorithms instead.

b) Test Problems: Two complementary test suites are employed in this study. Given the strong optimization capability of all selected comparison algorithms, the official SDC test suite from the CEC 2025 competition is adopted as the primary benchmark, consisting of 15 problems designed to evaluate algorithmic performance under complex constraint structures and difficult feasible regions. To maintain consistency in experimental scale and to assess practical applicability, 15 real-world constrained multi-objective problems are also selected from the RWMOP benchmark, drawn from representative engineering domains such as structural design, aerospace engineering, process systems, and power electronics. By combining a highly challenging competition benchmark with matched real-world problems, the experimental setup enables a comprehensive evaluation of both robustness and practical effectiveness. All experiments are conducted on the PlatEMO platform [26] to ensure a consistent implementation environment and reproducible experimental conditions. The experiments were performed on a Windows 11 (64-bit) operating system with an AMD Ryzen™ 9-7945HX (3.6 GHz) processor and 32 GB RAM, using MATLAB R2024b (Version 24.2.0.2712019) for all implementations.

TABLE I

MEAN (STANDARD DEVIATION) OF IGD AND HV OF THE COMPARED ALGORITHMS ON SDC PROBLEMS OVER 30 INDEPENDENT RUNS AT THE MAXIMUM OF 200,000 FUNCTION EVALUATIONS, WITH SIGNIFICANCE ASSESSED BY THE WILCOXON RANK-SUM TESTS ON IGD.

Prob	M	D	CLSHRDE		RDE		DESDE		IMTCMO		A-CCEMT	
			HV	IGD	HV	IGD	HV	IGD	HV	IGD	HV	IGD
F1	2	30	9.64E-02 (1.01E-01)	4.56E-01 (1.28E-01)	8.96E-02 (8.83E-02)	4.17E-01 + (1.72E-01)	**1.86E-01 (1.13E-01)**	**2.15E-01** - **(2.03E-01)**	1.11E-01 (9.67E-02)	4.31E-01 - (6.22E-01)	0.00E+00 (0.00E+00)	4.41E-01 + (1.45E-01)
F2	2	30	3.16E-01 (2.24E-02)	**2.17E-02 (2.10E-02)**	**3.19E-01 (1.67E-02)**	2.68E-02 + (1.60E-02)	3.14E-01 (1.39E-02)	2.56E-02 = (7.78E-03)	3.07E-01 (1.60E-02)	3.48E-02 + (1.17E-02)	1.63E-01 (5.55E-02)	2.35E-02 = (1.24E-02)
F3	2	30	**2.19E-01 (3.44E-02)**	**9.66E-01 (6.62E-02)**	2.16E-01 (2.66E-02)	9.95E-01 + (1.22E-01)	2.02E-01 (2.20E-02)	9.98E-01 + (1.27E-01)	1.41E-01 (2.51E-02)	1.08E+00 + (3.42E-02)	5.00E-03 (1.01E-02)	1.01E+00 + (3.66E-02)
F4	2	30	0.00E+00 (0.00E+00)	6.60E+00 (5.53E+01)	0.00E+00 (0.00E+00)	1.89E+01 = (1.13E+02)	0.00E+00 (0.00E+00)	**1.09E+00** = **(2.72E-02)**	0.00E+00 (0.00E+00)	1.11E+00 + (5.07E-02)	0.00E+00 (0.00E+00)	8.02E+01 = (4.16E+02)
F5	2	30	0.00E+00 (0.00E+00)	6.65E+01 (3.91E+01)	0.00E+00 (0.00E+00)	**6.00E+01** = **(3.36E+01)**	0.00E+00 (0.00E+00)	6.50E+01 = (2.93E+01)	0.00E+00 (0.00E+00)	1.11E+02 + (4.58E+01)	0.00E+00 (0.00E+00)	7.87E+01 + (4.12E+01)
F6	2	30	0.00E+00 (0.00E+00)	4.16E+02 (7.10E+02)	0.00E+00 (0.00E+00)	3.44E+02 - (7.05E+02)	0.00E+00 (0.00E+00)	4.78E+02 = (8.23E+02)	0.00E+00 (0.00E+00)	4.55E+02 = (5.25E+02)	0.00E+00 (0.00E+00)	**1.50E+02** - **(2.96E+02)**
F7	2	30	1.61E-01 (8.57E-02)	**1.49E-01 (8.86E-02)**	**1.62E-01 (7.81E-02)**	1.65E-01 = (7.83E-02)	9.66E-02 (5.37E-02)	2.46E-01 + (5.32E-02)	1.53E-01 (8.49E-02)	1.82E-01 + (9.23E-02)	2.57E-02 (2.79E-02)	1.73E-01 + (7.69E-02)
F8	2	30	9.09E-02 (1.55E-06)	7.31E-01 (7.74E-02)	9.55E-02 (2.50E-02)	7.42E-01 + (2.21E-04)	2.23E-01 (1.04E-01)	1.51E-01 - (2.04E-01)	**2.99E-01 (6.77E-02)**	**1.08E-01** - **(1.79E-01)**	1.07E-01 (5.05E-02)	7.42E-01 = (2.66E-06)
F9	2	30	0.00E+00 (0.00E+00)	**1.49E+01 (8.26E+00)**	0.00E+00 (0.00E+00)	1.98E+01 + (1.46E+01)	0.00E+00 (0.00E+00)	2.91E+01 + (1.98E+01)	0.00E+00 (0.00E+00)	3.29E+02 + (1.02E+02)	0.00E+00 (0.00E+00)	2.06E+01 + (1.94E+01)
F10	2	30	0.00E+00 (0.00E+00)	6.43E+00 (1.97E+00)	0.00E+00 (0.00E+00)	**6.24E+00** = **(2.03E+00)**	0.00E+00 (0.00E+00)	6.77E+00 + (1.52E+00)	0.00E+00 (0.00E+00)	7.50E+00 + (1.47E+00)	0.00E+00 (0.00E+00)	6.36E+00 = (1.93E+00)
F11	2	30	**1.52E-01 (1.87E-01)**	**5.69E+00 (5.47E+00)**	8.78E-02 (1.46E-01)	5.99E+00 = (5.40E+00)	1.68E-02 (6.01E-02)	7.57E+00 + (5.54E+00)	4.53E-04 (2.48E-03)	9.64E+00 + (2.62E+00)	0.00E+00 (0.00E+00)	7.59E+00 = (7.19E+00)
F12	2	30	8.51E-02 (2.31E-02)	8.27E-01 (2.24E-01)	9.11E-02 (2.89E-04)	8.05E-01 + (4.02E-04)	8.42E-02 (7.34E-02)	1.29E+00 = (1.51E+00)	**1.82E-01 (1.59E-01)**	**5.48E-01** - **(8.25E-01)**	3.85E-02 (3.45E-02)	8.63E-01 = (3.18E-01)
F13	2	30	0.00E+00 (0.00E+00)	**7.15E+00 (2.36E+00)**	0.00E+00 (0.00E+00)	7.64E+00 + (2.42E+00)	0.00E+00 (0.00E+00)	8.92E+00 + (1.62E+00)	0.00E+00 (0.00E+00)	1.39E+01 + (2.21E+00)	0.00E+00 (0.00E+00)	7.51E+00 = (2.11E+00)
F14	2	30	0.00E+00 (0.00E+00)	**7.38E+01 (4.96E+01)**	0.00E+00 (0.00E+00)	9.00E+01 + (6.80E+01)	0.00E+00 (0.00E+00)	1.20E+02 + (5.41E+01)	0.00E+00 (0.00E+00)	6.78E+02 + (3.96E+02)	0.00E+00 (0.00E+00)	1.06E+02 + (7.08E+01)
F15	3	30	**5.26E-01 (6.45E-03)**	**6.46E-02 (2.34E-03)**	5.19E-01 (7.68E-03)	7.18E-02 + (4.47E-03)	4.93E-01 (2.00E-02)	8.99E-02 + (9.11E-03)	4.10E-01 (1.10E-01)	1.11E+00 + (3.03E+00)	3.13E-02 (8.03E-02)	7.21E-02 + (5.54E-03)
W/T/L			–/–/–		9/5/1		8/5/2		11/1/3		7/7/1	

c) Parameter setup: A unified parameter configuration is adopted for all experiments. The population size of each algorithm is fixed at 100. The maximum number of function evaluations is set to 200,000, and the decision variable dimension is uniformly set to 30 for all test problems. Each algorithm is executed independently for 30 runs on both the SDC test suite and the RWMOP problems, following common experimental practice in constrained multi-objective optimization studies [26], [27]. This configuration provides sufficient evolutionary budget and population diversity for advanced CMOP algorithms, while maintaining a balanced trade-off between computational cost and evaluation reliability. The additional computational overhead is negligible, as the proposed method does not introduce extra function evaluations.

d) Performance indicators: To quantitatively evaluate the performance of the compared algorithms, three widely used performance indicators are adopted, namely the mean inverted generational distance (IGD), the mean hypervolume (HV) [15], and the feasible success rate (FSR) [28]. FSR is defined as the proportion of runs that produce at least one feasible solution in the final population. All pairwise statistical comparisons are performed at $\alpha = 0.05$ using one-sided Wilcoxon rank-sum tests on the run-wise IGD values, and the outcomes are encoded by $+$, $=$, and $-$. If no feasible solution is obtained at termination in a run, the resulting IGD and HV outcomes are not comparable; therefore, the corresponding entry is reported as $\mathrm{NaN}(\mathrm{FSR})$, where the number in parentheses indicates the FSR over 30 independent runs. For statistical testing, infeasible runs are treated as the worst case by assigning a large penalty value to the corresponding indicator value. Specifically, any Not a Number (NaN) IGD is replaced by a penalty P defined per problem as a multiple of the largest finite IGD observed among both algorithms, e.g., $P = 10 \times \max(\mathcal{V})$, where $\mathcal{V}$ contains all non-NaN IGD values from both algorithms on the same problem (a fixed large fallback is used if $\mathcal{V}$ is empty) [29]. The Wilcoxon rank-sum tests is then applied to the penalized run-wise IGD values, and the resulting $+/=/-$ symbols are reported as the Wilcoxon test results.

B. Experimental results

Table I summarizes the results of the experimental comparison between the proposed algorithm and other competitive methods on the SDC test suite. The table reports the mean and standard deviation of the IGD and HV metrics obtained over multiple independent runs for each problem.

It should be noted that all the algorithms compared demonstrate a strong optimization capability in SDC problems and achieve a FSR of 100%. This indicates that the selected methods are equipped with effective constraint-handling mechanisms and are able to consistently locate feasible regions even under highly constrained scenarios. However, due to the extremely strict constraint settings and narrow feasible regions of the SDC test suite, the HV values of multiple algorithms are zero on many problems. This phenomenon does not imply optimization failure; rather, it reflects the substantial difficulty of obtaining a well-distributed feasible approximation of the constrained Pareto front under such

TABLE II
MEAN (STANDARD DEVIATION) OF IGD AND HV OF THE COMPARED ALGORITHMS ON RWMOP PROBLEMS OVER 30 INDEPENDENT RUNS AT THE MAXIMUM OF 200,000 FUNCTION EVALUATIONS, WITH SIGNIFICANCE ASSESSED BY THE WILCOXON RANK-SUM TESTS ON IGD.

Prob	M	D	CLSHRDE		RDE		DESDE		IMTCMO		A-CCEMT	
			HV (SD)	IGD (SD)	HV (SD)	IGD (SD)	HV (SD)	IGD (SD)	HV (SD)	IGD (SD)	HV (SD)	IGD (SD)
F1	2	4	6.08E-01 (3.70E-04)	3.55E+05 (2.02E+03)	3.06E-01 (1.04E-01)	**3.52E+05** - **(5.73E+03)**	6.08E-01 (3.27E-04)	3.55E+05 = (2.76E+03)	6.08E-01 (3.62E-04)	3.59E+05 + (1.10E+03)	**6.08E-01** **(2.50E-04)**	3.58E+05 + (1.30E+03)
F2	2	5	3.93E-01 (1.72E-05)	**6.68E-01** **(6.86E-01)**	1.85E-01 (6.31E-02)	5.48E+00 + (5.32E+00)	3.93E-01 (1.38E-05)	6.93E-01 = (6.39E-01)	**3.93E-01** **(1.26E-05)**	6.22E+00 + (7.98E+00)	3.93E-01 (1.38E-05)	8.92E+00 + (1.34E+01)
F3	2	3	8.99E-01 (6.29E-04)	3.24E+01 (3.06E+01)	1.95E-01 (8.13E-02)	9.97E+04 + (1.08E+03)	8.99E-01 (5.10E-04)	4.88E+01 = (4.46E+01)	8.99E-01 (7.60E-04)	2.04E+01 = (1.81E+01)	**9.00E-01** **(4.95E-04)**	**1.22E+01** - **(9.88E+00)**
F4	2	4	8.25E-01 (1.26E-02)	2.09E-01 (2.82E-01)	3.35E-01 (1.79E-01)	3.04E-01 = (6.15E-01)	8.57E-01 (3.60E-03)	2.93E-01 = (3.48E-01)	8.60E-01 (8.60E-04)	2.31E-01 = (2.23E-01)	**8.60E-01** **(8.10E-04)**	**8.94E-02** = **(7.32E-02)**
F5	2	4	4.35E-01 (1.22E-04)	1.77E+00 (9.08E-02)	2.61E-01 (8.44E-02)	1.39E+00 - (2.48E-01)	**4.35E-01** **(9.06E-05)**	1.75E+00 = (1.18E-01)	4.35E-01 (9.63E-05)	**1.65E+00** - **(1.35E-01)**	4.35E-01 (9.95E-05)	1.73E+00 - (8.73E-02)
F6	2	4	4.85E-01 (6.24E-05)	2.94E+00 (1.26E+00)	9.17E-01 (4.61E-02)	3.52E+00 = (1.92E+00)	4.85E-01 (4.75E-05)	3.05E+00 = (1.34E+00)	4.85E-01 (4.23E-05)	2.68E+00 = (1.00E+00)	**4.85E-01** **(3.46E-05)**	**2.64E+00** = **(9.59E-01)**
F7	2	4	4.09E-01 (1.46E-04)	**3.72E-02** **(1.41E-17)**	3.97E-01 (1.35E-01)	9.59E+00 + (2.81E+01)	4.10E-01 (1.21E-04)	3.72E-02 - (5.04E-08)	4.10E-01 (1.44E-04)	3.72E-02 - (4.80E-08)	**4.10E-01** **(1.20E-04)**	**3.72E-02** = **(1.41E-17)**
F8	2	2	8.42E-01 (1.67E-03)	1.52E-03 (1.27E-03)	2.32E-01 (2.09E-01)	2.37E-01 + (3.76E-01)	8.42E-01 (1.92E-03)	1.35E-03 = (1.16E-03)	**8.42E-01** **(1.41E-03)**	1.40E-03 = (1.44E-03)	8.41E-01 (2.25E-03)	**1.06E-03** = **(9.84E-04)**
F9	2	4	**5.56E-01** **(1.16E-03)**	6.23E-02 (1.60E-02)	2.01E-01 (3.34E-02)	5.95E-01 + (9.66E-01)	5.56E-01 (1.68E-03)	6.03E-02 = (1.37E-02)	5.55E-01 (1.91E-03)	5.74E-02 = (1.13E-02)	5.55E-01 (1.43E-03)	**5.62E-02** = **(7.10E-03)**
F10	3	7	8.74E-02 (2.10E-04)	3.65E+02 (2.75E+00)	**3.02E-01** **(2.08E-01)**	5.17E+02 + (3.03E+02)	8.75E-02 (1.67E-04)	3.65E+02 = (2.42E+00)	8.75E-02 (1.55E-04)	3.65E+02 = (2.26E+00)	8.73E-02 (1.63E-04)	**3.64E+02** = **(3.16E+00)**
F11	2	3	5.44E-01 (3.09E-05)	**5.36E+01** **(3.30E+01)**	5.44E-01 (4.16E-05)	7.33E+01 = (4.56E+01)	5.44E-01 (2.71E-05)	5.40E+01 = (3.60E+01)	**5.44E-01** **(1.67E-05)**	6.58E+01 = (6.92E+01)	5.44E-01 (2.70E-05)	7.38E+01 = (6.72E+01)
F12	3	6	4.05E-01 (1.79E-02)	**1.07E+03** **(3.63E+02)**	1.12E-01 (1.05E-01)	2.79E+03 + (6.78E+02)	**4.09E-01** **(2.19E-04)**	2.36E+03 + (9.11E+02)	3.88E-01 (2.58E-02)	2.38E+03 + (7.90E+02)	4.03E-01 (2.73E-02)	2.27E+03 + (7.55E+02)
F13	2	6	3.18E-02 (9.61E-07)	1.61E-02 (1.29E-09)	**9.37E-02** **(1.16E-03)**	1.99E-02 + (4.07E-03)	3.18E-02 (1.17E-06)	**1.61E-02** = **(1.06E-17)**	3.18E-02 (1.12E-06)	**1.61E-02** = **(1.06E-17)**	3.18E-02 (9.38E-07)	1.61E-02 - (2.44E-08)
F14	3	9	8.97E-01 (2.71E-01)	5.46E+00 (2.41E+00)	1.00E-01 (3.05E-01)	6.77E+03 + (3.70E+04)	7.57E-01 (3.89E-01)	**1.81E+00** - **(1.88E+00)**	9.88E-01 (6.44E-02)	NaN(97%) + (–)	**1.00E+00** **(2.10E-17)**	NaN(97%) + (–)
F15	2	25	4.51E-01 (3.17E-01)	**5.32E-02** **(5.03E-02)**	**8.05E-01** **(9.53E-02)**	NaN(97%) + (–)	7.72E-01 (1.36E-01)	NaN(77%) + (–)	7.80E-01 (1.18E-01)	NaN(53%) + (–)	3.26E-01 (2.77E-01)	NaN(57%) + (–)
W/T/L			-/-/-		10/3/2		2/11/2		5/8/2		5/7/3	

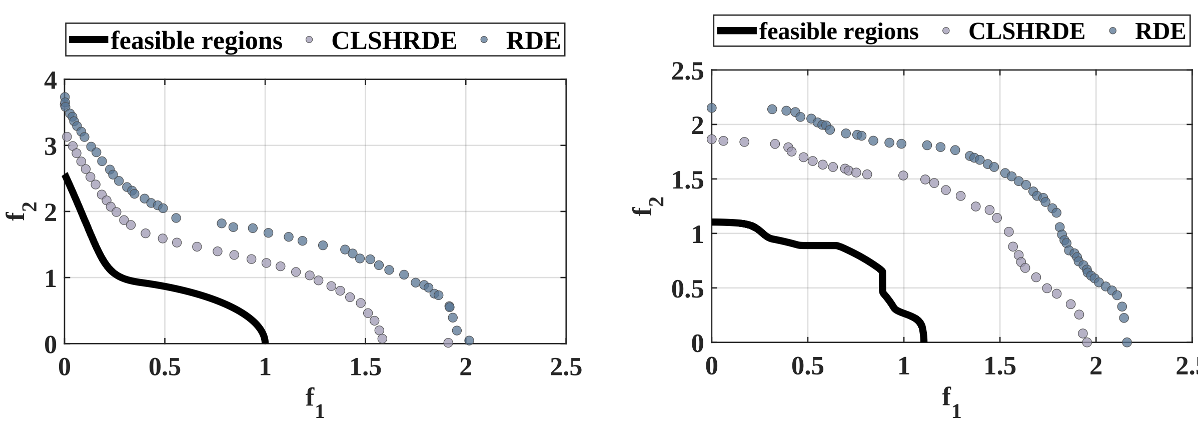

Fig. 1. Pareto fronts on SDC problems F3 and F4 obtained by CLSHRDE and RDE over 30 independent runs in the (f_1, f_2) objective space.

challenging conditions. Although some test instances exhibit relatively large standard deviations, the Wilcoxon rank-sum test remains a standard nonparametric tool for comparing the distributions of independent runs in evolutionary optimization. Moreover, the reported results are supported jointly by IGD, HV, and FSR, which provides complementary evidence for the observed performance differences.

Overall, the results in Table I show that although the SDC benchmark poses severe challenges in terms of constraint satisfaction and feasible-region exploration, the compared algorithms are still able to maintain high feasibility performance, providing a rigorous and discriminative testbed for evaluating advanced constrained multi-objective optimization methods. Further analysis reveals that the proposed method exhibits statistically significant advantages on problems F2, F3, F7, F9, F11, F13, F14, and F15, while maintaining performance comparable to mainstream algorithms on the remaining problems, demonstrating its stability and balanced behavior across different levels of constraint difficulty.

From the perspective of the Pareto front distributions shown in Fig. 1, CLSHRDE exhibits a clear advantage over RDE on both SDC-F3 and SDC-F4. The solutions obtained by

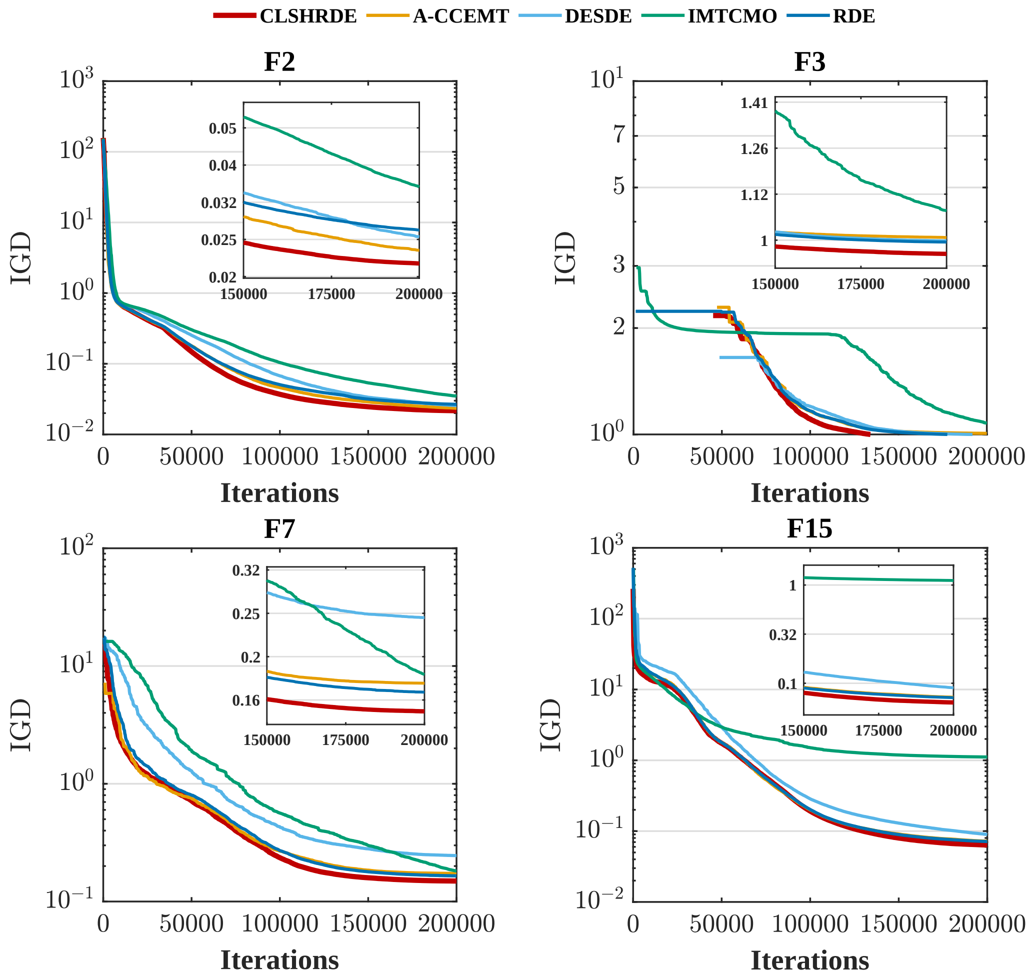

Fig. 2. Mean IGD convergence curves ($\log_{10}$) on SDC problems, averaged over 30 independent runs.

CLSHRDE are consistently closer to the feasible Pareto front boundary in the (f_1, f_2) objective space and form a more compact and well-aligned approximation, indicating superior convergence behavior, where the symbols f_1 and f_2 in Fig. 1 represent the objective function values evaluated at the obtained Pareto-optimal solutions, following the standard notation in multi-objective optimization.

In contrast, the Pareto fronts generated by RDE are visibly farther from the feasible boundary and display a looser distribution, suggesting insufficient exploitation of the constrained feasible region. Moreover, CLSHRDE achieves a smoother and more continuous front shape, reflecting its stronger capability in navigating narrow feasible regions and maintaining effective search guidance under strict constraints.

Fig. 2 presents the IGD convergence curves of the compared algorithms on SDC-F2, F3, F7, and F15 throughout the optimization process. While the algorithms exhibit broadly similar convergence trends in the early stage, CLSHRDE achieves and maintains lower IGD values in the later stage across all four problems, indicating more stable late-stage convergence and improved final solution quality under strict constraint conditions. Each curve is obtained by averaging the IGD values over 30 independent runs at each iteration, which explains the relatively smooth appearance. Fig. 2 is intended to visualize the overall convergence trends, so we present the mean curves only. The dispersion across 30 independent runs is quantified separately through run-wise statistical testing and summarized in the corresponding tables, providing a rigorous complement to the trend plot.

Table II summarizes the experimental results of all algorithms on the RWMOP test suite. Compared with the SDC benchmark, the RWMOP problems exhibit more complex and irregular feasible regions, which leads to frequent occurrences of NaN results on problems F14 and F15 for multiple algorithms. Notably, on the most challenging case F15, CLSHRDE is the only algorithm that achieves a 100% feasibility rate, while all other methods suffer from infeasible runs and are consequently marked as NaN. This result can be attributed to the stage-aware search mechanism of CLSHRDE, which explicitly distinguishes different phases of the evolutionary process and adaptively balances exploration and exploitation across these stages. By gradually shifting the search emphasis from feasibility establishment to solution refinement, the proposed approach enhances its capability of navigating highly constrained and structurally complex feasible regions, leading to consistently feasible search behavior on challenging real-world problems.

In conclusion, the experimental studies on both benchmark and real-world test suites confirm the effectiveness of the proposed approach for constrained multi-objective optimization problems. The results indicate that the method can reliably handle complex constraints while maintaining competitive solution quality across different problem settings.

V. CONCLUSION

In this paper, we propose CLSHRDE, a novel differential evolution framework that integrates a two-operator, success-driven competition mechanism with a closed-loop exploration–exploitation scheduling strategy. By jointly leveraging success-rate feedback and smooth scheduling, CLSHRDE adaptively balances operator selection and achieves stable and accurate feasible-front approximation. Notably, CLSHRDE is the only method achieving full feasibility on the challenging RWMOP-F15 problem.Extensive experiments on the SDC and RWMOP benchmark suites demonstrate that the proposed method consistently outperforms representative state-of-the-art algorithms in terms of convergence accuracy, feasibility maintenance, and robustness under a unified parameter configuration. Moreover, the exponentially weighted moving average employed in CLSHRDE can be viewed as a simplified instance of memory mechanisms related to fractional-order difference models. This perspective suggests that extending the current framework toward fractional-order memory modeling may enable richer temporal dynamics and smoother success accumulation, constituting a promising direction for future research.

REFERENCES

[1] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[2] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: A review of algorithms and comparison with derivative-based methods," *Journal of Global Optimization*, vol. 56, pp. 1247–1299, 2013.

[3] J. Liang, H. Lin, C. Yue, X. Ban, and K. Yu, "Evolutionary constrained multi-objective optimization: a review," *Vicinagearth*, vol. 1, p. 5, 2024.

[4] T. Tušar, A. Vodopija, and B. Filipič, "Visual exploration of the effect of constraint handling in multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2023, pp. 3–16.

[5] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert systems with applications*, vol. 123, pp. 108–126, 2019.

[6] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 2419–2426.

[7] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, "Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9559–9572, 2021.

[8] 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.

[9] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 201–221, 2022.

[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] CEC 2025 CMOP Competition Committee, "CEC 2025 Competition on Constrained Multiobjective Optimization Problems (CMOP)," Competition technical report and benchmark definitions, 2025.

[12] 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.

[13] K. Qiao, J. Liang, K. Yu, C. Yue, H. Lin, D. Zhang, and B. Qu, "Evolutionary constrained multiobjective optimization: Scalable high-dimensional constraint benchmarks and algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 4, pp. 965–979, 2023.

[14] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das, "A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results," *Swarm and Evolutionary Computation*, vol. 67, p. 100961, 2021.

[15] Y. Yang, J. Liu, and S. Tan, "A constrained multi-objective evolutionary algorithm based on decomposition and dynamic constraint-handling mechanism," *Applied Soft Computing*, vol. 89, p. 106104, 2020.

[16] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2018.

[17] 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.

[18] 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)*, 2014, pp. 1658–1665.

[19] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing shade and lshade algorithms for global numerical optimization," *Swarm and Evolutionary Computation*, vol. 50, p. 100455, 2019.

[20] B. Billah, M. L. King, R. D. Snyder, and A. B. Koehler, "Exponential smoothing model selection for forecasting," *International journal of forecasting*, vol. 22, no. 2, pp. 239–247, 2006.

[21] V. E. Tarasov and V. V. Tarasova, "Long and short memory in economics: fractional-order difference and differentiation," *arXiv preprint arXiv:1612.07903*, 2016.

[22] V. Minzu, S. Riahi, and E. Rusu, "Implementation aspects regarding closed-loop control systems using evolutionary algorithms," *Inventions*, vol. 6, no. 3, p. 53, 2021.

[23] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM computing surveys (CSUR)*, vol. 45, no. 3, pp. 1–33, 2013.

[24] X. Ban, J. Liang, K. Yu, K. Qiao, Y. Wang, and J. Peng, "A dynamic exemplars selection-based differential evolution algorithm for constrained multi-objective optimization," in *2024 7th International Symposium on Autonomous Systems (ISAS)*. IEEE, 2024, pp. 1–8.

[25] S. Liu, Z. Wang, Q. Lin, and J. Chen, "Coevolutionary multitasking for constrained multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 91, p. 101727, 2024.

[26] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[27] K. Qiao, J. Liang, K. Yu, W. Guo, C. Yue, B. Qu, and P. N. Suganthan, "Benchmark problems for large-scale constrained multi-objective optimization with baseline results," *Swarm and Evolutionary Computation*, vol. 86, p. 101504, 2024.

[28] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[29] 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.