Multiple Mutation Strategy Differential Evolution with Allocation of Elite Individuals to the Best-Performing Strategy

Authors: Dongfang Zhang

Date: 2024-12-10T00:00:00+00:00

Abstract

Real parameter single objective optimization has been a prominent field for these decades. Recently, long-term search of real parameter single objective optimization is widely concerned based on the fact that solving difficulty always scales exponentially with the increase of dimensionality of solution space. So far, a number of population-based metaheuristics have been proposed. Among the algorithms, IMODE - a differential evolution algorithm based on three mutation strategies and the binomial or exponential crossover - demonstrates good performance. In this paper, based on IMODE, we propose multiple mutation strategies Differential Evolution with the Best Individuals allocated to the Best performer among the Strategies - BIBSDE - by revising IMODE. Altogether, we make five revisions in algorithm behavior and a change in parameter setting. The most important revision is that, during execution, for the next generation, the current best individuals are allocated to the best performer among the three mutation strategies as reward. Experimental results show that our BIBSDE performs better or at least not worse than existing population based metaheuristics for long-term search. Besides, each measure proposed by us is effective for enhancement.

Full Text

Preamble

Multiple Mutation Strategies Differential Evolution With the Best Individuals Allocated to the Best Performer Among the Strategies

Chengjun Li[a,b], Hongbo Mei[c,*], Fanyu Liu[a], Mingyuan Bai[d], Mukang Miao[a], Wenxuan Yang[a], Jinghan Hu[a], Dongfang Zhang[a], Lei Kuang[a]

[a]School of Computer Science, China University of Geosciences, Wuhan 430078, P.R. China
[b]Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430078, P.R. China
[c]School of Earth Resources, China University of Geosciences, Wuhan 430078, P.R. China
[d]RIKEN AIP, Tokyo 1030027, Japan

Abstract

Real-parameter single-objective optimization has been a prominent research area for decades. Recently, long-term search for real-parameter single-objective optimization has attracted widespread concern, as solving difficulty typically scales exponentially with increasing dimensionality of the solution space. Numerous population-based metaheuristics have been proposed to address this challenge. Among these, IMODE—a differential evolution algorithm based on three mutation strategies and binomial or exponential crossover—demonstrates strong performance. In this paper, we propose BIBSDE (Multiple Mutation Strategies Differential Evolution with the Best Individuals allocated to the Best performer among the Strategies), a revised version of IMODE. We introduce five algorithmic modifications and one parameter adjustment. The most significant revision allocates current best individuals to the best-performing mutation strategy as a reward for the next generation. Experimental results demonstrate that BIBSDE performs better than, or at least not worse than, existing population-based metaheuristics for long-term search, and each proposed measure contributes effectively to performance enhancement.

Keywords: differential evolution, ensemble, allocation, reward, long-term search

2010 MSC: 00-01, 99-00

Corresponding author: hbmei cug@163.com

Preprint submitted to Information Sciences
December 10, 2024

1. Introduction

Real-parameter single-objective optimization involves minimizing or maximizing an objective function by finding the optimal decision vector within the solution space. Over the years, various population-based metaheuristics have been proposed for this purpose, among which Differential Evolution (DE) has shown remarkable performance. For instance, in the influential competitions on real-parameter single-objective optimization held at the Congress on Evolutionary Computation (CEC), most winners have been DE-based algorithms.

DE operates through three main operators: mutation, crossover, and selection. Initially, individuals in the population—target vectors $\vec{x}_{i,0} = (x_{1,i,0}, x_{2,i,0}, ..., x_{D,i,0})$, where $i \in \{1, 2, ..., NP\}$, NP denotes population size, and D represents dimensionality—are generated, typically through random initialization. Evolution then proceeds across generations. In the gth generation, mutant vectors $\vec{v}_{i,g}$ are generated through mutation. Numerous mutation strategies have been proposed in the literature. For example, the early DE/best/1 strategy is given by:

$$\vec{v}_{i,g} = \vec{x}_{best,g} + F \cdot (\vec{x}_{r1,g} - \vec{x}_{r2,g}) \tag{1}$$

where $\vec{x}_{best,g}$ denotes the best target vector in the gth generation, $r1, r2 \in \{1, 2, ..., NP\}$ are randomly generated with $r1 \neq r2 \neq i$, and F is the scaling factor parameter.

Building upon early strategies, more complex variants have been developed, such as DE/current-to-ϕbest/1:

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F_i \cdot (\vec{x}_{\phi,g} - \vec{x}_{i,g} + \vec{x}_{r1,g} - \vec{x}_{r2,g}) \tag{2}$$

DE/current-to-ϕbest/1 with archive:

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F_i \cdot (\vec{x}_{\phi,g} - \vec{x}_{i,g} + \vec{x}_{r1,g} - \vec{x}_{r3,g}) \tag{3}$$

and DE/weighted-rand-to-ϕbest/1:

$$\vec{v}_{i,g} = F_i \cdot \vec{x}_{r1,g} + (\vec{x}_{\phi,g} - \vec{x}_{r2,g}) \tag{4}$$

In Equations 2–4, $r1 \neq r2 \neq r3 \neq i$, with $r1, r2 \in \{1, 2, ..., NP\}$ and $r3 \in \{1, 2, ..., NP, ..., NP + |A|\}$, where $|A|$ is the size of an external archive collecting target vectors eliminated during selection. Thus, $\vec{x}_{r1,g}$ and $\vec{x}_{r2,g}$ are randomly chosen from the current population, while $\vec{x}_{r3,g}$ is chosen from either the population or the external archive. Additionally, $\vec{x}_{\phi,g}$ denotes a target vector among the $NP \cdot \phi$ best individuals in the gth generation, where $\phi \in (0, 1)$. Each mutant vector $\vec{v}_{i,g}$ has its own scaling factor F_i.

After mutation, trial vectors $\vec{u}_{i,g} = (u_{1,i,g}, u_{2,i,g}, ..., u_{D,i,g})$ are generated from $\vec{x}_{i,g}$ and $\vec{v}_{i,g}$ through crossover. The binomial crossover, defined as:

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } \mathrm{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{j,i,g}, & \text{otherwise} \end{cases} \tag{5}$$

is most commonly used, where $CR \in [0, 1]$ is the crossover rate and j_{rand} is a randomly generated integer from $[1, D]$ ensuring at least one component comes from $\vec{v}_{i,g}$. Alternative strategies exist, such as the binomial or exponential crossover:

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } \mathrm{rand}(0,1) \leq CR \text{ or } j = \mathrm{randn}(i) \\ x_{j,i,g}, & \text{otherwise} \end{cases} \tag{6}$$

with probability p for the binomial manner (and $1-p$ for exponential), where $l \in \{1, 2, ..., D\}$ is randomly determined during execution and $L \in \{1, 2, ..., D\}$ is a preset parameter. Mutation and crossover are collectively termed the trial vector generation strategy.

For selection:

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g}, & \text{if } f(\vec{u}_{i,g}) \leq f(\vec{x}_{i,g}) \\ \vec{x}_{i,g}, & \text{otherwise} \end{cases} \tag{7}$$

where $f(\vec{u}_{i,g})$ and $f(\vec{x}_{i,g})$ represent fitness values obtained through function evaluation.

To enhance DE performance for real-parameter single-objective optimization, researchers have pursued at least four directions: (1) innovation of trial vector generation strategies; (2) adaptation of control parameter settings; (3) hybridization with non-DE search techniques; and (4) ensemble of multiple trial vector generation strategies or even DE algorithms.

Research in real-parameter single-objective optimization has focused on two search types. Traditional search, where the maximum number of function evaluations ($MaxFES$) scales linearly with dimensionality ($MaxFES = D \cdot 1.00E{+}04$ for $D \in \{10, 30, 50, 100\}$), has been studied for decades and was the focus of CEC competitions before 2020. However, solving difficulty often scales exponentially with D, motivating recent interest in long-term search, where $MaxFES$ scales linearly with dimensionality. The CEC 2020, 2021, and 2022 competitions focused on long-term search with D ranging from 10 to 20, $MaxFES$ ranging from $1.00E + 06$ to $1.00E + 07$, and performance showing less variation with increasing D.

Since 2020, population-based metaheuristics suitable for long-term search have been proposed, featuring measures to slow convergence velocity and enable broader search to resist stagnation. Among these, the Improved Multi-Operator DE algorithm (IMODE) [16], an ensemble of multiple trial vector generation strategies using the three mutation strategies from Equations 2–4 and the binomial or exponential crossover from Equation 6, was the top performer in CEC 2020. In each generation, IMODE computes the average improvement Δf_k from target to trial vectors for positions controlled by the kth mutation strategy, then calculates the controlling ratio c_k for the next generation as:

$$c_k = \frac{\sum_{i=1}^{3} \Delta f_i}{\Delta f_k} \tag{8}$$

Better performance yields larger controlling ratios, with a constraint $0.1 \leq c_k \leq 0.9$ enforced by adjusting c_{max} if any $c_m < 0.1$. IMODE adapts F and CR using Cauchy and normal distributions respectively, following [22], and maintains an external archive. For selection, Equation 7 is used. In each generation, $c_k \cdot NP$ individuals are randomly allocated to the kth mutation strategy, and in the final stage, a sequential quadratic programming-based local search is executed.

In IMODE, individuals may be allocated to any mutation strategy without restriction, making allocation completely random. This slows convergence and reduces stagnation probability, making it suitable for long-term search. However, this excessively random scheme may not be optimal. To improve IMODE, we propose reducing allocation randomness by rewarding the best-performing mutation strategy with the best individuals, among other adjustments.

2. Related Work

This section reviews existing algorithms for long-term search in real-parameter single-objective optimization. While fewer than traditional search algorithms, their number is increasing rapidly.

The top three algorithms in CEC 2020 were IMODE [16], AGSK [23], and j2020 [1]. IMODE uses three mutation strategies competing for individuals and binomial/exponential crossover. AGSK is an adaptive version of the gaining-sharing knowledge algorithm, executing junior and senior phases on different dimensions with adaptively controlled rates. j2020 [1], based on the famous self-adaptive jDE [28] and its two-subpopulation variant jDE100 [29], incorporates crowding and mutation individual selection from both subpopulations.

The CEC 2021 suite parameterizes functions with bias, rotation, and translation, requiring separate rankings for non-shifted, shifted, non-rotated shifted, and rotated shifted cases. APGSK-IMODE [14] ranked first for non-shifted cases, jDE-21 [3] for non-rotated shifted cases, and NL-SHADE-RSP [5] for both shifted and rotated shifted cases. APGSK-IMODE ensembles AGSK and IMODE, with each controlling a subpopulation and sharing the best individual periodically. jDE-21 extends j2020 with a restart mechanism and revised crowding/mutation selection. NL-SHADE-RSP refines L-SHADE-RSP [30] by tuning selective pressure and adding automatic archive usage probability tuning.

The top three CEC 2022 algorithms were EA4eig [15], NL-SHADE-LBC [9], and NL-SHADE-RSP-MID [8]. EA4eig ensembles CoBiDE [31], IDEbd [32], CMA-ES [33], and jSO [34], extending CoBiDE's Eigen approach to all components. NL-SHADE-LBC integrates selective pressure, parameter adaptation with linear bias change, current-to-pbest mutation, bound constraint resampling, and NLPSR. NL-SHADE-RSP-MID enhances NL-SHADE-RSP with midpoint-based best result updating, restart triggering, and k-means population clustering.

Recent work includes APGSK-IMODE-FL [21], which revises APGSK-IMODE with a more complex exchange scheme based on lifetime and fitness, and AM-CDE [20], which alternates between monopoly (one strategy controls all individuals) and competition (three strategies compete for control) states. Population reduction is widely used: LPSR in IMODE and j2020, and NLPSR in others. NLPSR, defined by:

$$NP = \text{round}\left(\left(\frac{MaxFES - FES}{MaxFES}\right)^{1-\frac{FES}{MaxFES}} \cdot (NP_{min} - NP_{max}) + NP_{max}\right) \tag{13}$$

where FES is consumed evaluations and NP_{max}, NP_{min} are initial and final population sizes, decreases rapidly early and slowly later. Pre-experimental analysis suggests NLPSR may outperform LPSR for long-term search.

3. Methodology

While IMODE performs well, further improvements are possible. We focus on three aspects: population size reduction, trial vector generation parameter setting, and individual allocation to mutation strategies.

Population Size Reduction: Most long-term search metaheuristics use NLPSR. We replace LPSR with NLPSR and propose reducing individuals just better than the worst rather than the worst themselves. Let NP_{before} and NP_{after} be population sizes before and after reduction. Instead of removing the worst $NP_{before} - NP_{after}$ individuals, we remove those just better than them, preserving the worst individuals. However, if the best individual falls among those just better than the worst at the final evolutionary stage, the worst individuals are removed instead.

This approach maintains diversity by preserving individuals most different from the best, as worst individuals often exhibit maximal divergence. Since selection (Equation 7) drives convergence toward the best, this strategy reduces diversity loss while preserving good fitness individuals.

Parameter Setting: We emphasize adaptive parameter control. First, the best individual ratio ϕ for mutation strategies (Equations 2–4) descends linearly:

$$\phi = (\phi_{min} - \phi_{max}) \cdot \frac{MaxFES}{FES} + \phi_{max} \tag{14}$$

Second, CR adaptation is revised differently for execution halves. Following [5], where NL-SHADE-RSP's binomial crossover rate CR_b is:

$$CR_b = \begin{cases} 2 \cdot \frac{FES-0.5}{MaxFES}, & \text{if } FES \leq MaxFES \cdot 0.5 \\ \text{otherwise} \end{cases} \tag{15}$$

Our algorithm uses the original CR setting (Equations 11–12) in the first half, but uniformly sets and linearly descends CR in the second half:

$$CR = 2 \cdot \left(1 - \frac{MaxFES}{FES}\right) \tag{16}$$

Allocation Scheme: We implement a reward mechanism. While controlling ratios c_k are still computed via IMODE's method (Equation 8), allocation proceeds as: best individuals are assigned to the best-performing mutation strategy (highest c_k), while remaining individuals are randomly allocated to other strategies to maintain $c_k \cdot NP$ individuals per strategy. This reduces allocation randomness while rewarding superior performance.

Algorithm 1 details BIBSDE, which adds parameters ϕ_{max} and ϕ_{min} to IMODE. Key changes include revised CR setting (Steps 7–11), linear ϕ decrease (Step 12), identification of the best mutation strategy (Step 20), and allocation of best individuals to it (Step 21).

4. Experimental Study

The first experiment compares BIBSDE with seven algorithms from Section 2 (IMODE, AGSK, APGSK-IMODE, NL-SHADE-RSP, MLS-L-SHADE, EA4eig, AMCDE) on CEC 2020 and 2022 benchmark suites. Tables 1–2 summarize these suites. From CEC 2011 real-world problems, we select those with $D \leq 20$ (Table 3), adjusting $MaxFES$ for long-term search as shown in Table 4 . The third experiment verifies revision effectiveness on CEC 2020 with $D = 15$. All algorithms run 30 times per case.

4.1. Experimental Settings

Algorithm parameters are listed in Table 5 . BIBSDE retains most IMODE parameters but resets $p = 0.7$ for improvement, with $\phi_{max} = 0.4$ and $\phi_{min} = 0.2$ as new parameters.

4.2. Comparison Based on the CEC 2020 and 2022 Benchmark Test Suites

BIBSDE is compared with seven peers on CEC 2020 and 2022 suites at $D = 10$ and $D = 20$. Tables 6–9 present results with Wilcoxon rank sum test at 0.05 significance, while Tables 10–13 show Friedman test outcomes.

For CEC 2020 at $D = 10$, BIBSDE defeats all peers in Wilcoxon tests (Table 6) and ranks first in Friedman test (Table 10). At $D = 20$, BIBSDE is defeated by

NL-SHADE-RSP but outperforms others in Wilcoxon tests (Table 7), ranking second behind NL-SHADE-RSP in Friedman test (Table 11).

For CEC 2022 at $D = 10$, BIBSDE is defeated by EA4eig but outperforms others (Table 8), ranking first in Friedman test (Table 12). At $D = 20$, BIBSDE is defeated by MLS-L-SHADE but outperforms others (Table 9), ranking second behind MLS-L-SHADE (Table 13).

Summary comparisons appear in Tables 14 (Wilcoxon) and 15 (Friedman). Overall, BIBSDE demonstrates superior performance. Convergence graphs for selected CEC 2022 functions where global optima are not found are shown in Figures 1 [Figure 1: see original paper] ($D = 10$) and 2 [Figure 2: see original paper] ($D = 20$). BIBSDE shows significantly sharper convergence than IMODE in six cases, with non-significant differences in five others.

4.3. Comparison Based on Selected CEC 2011 Real-World Problems

BIBSDE is compared with IMODE, NL-SHADE-RSP, EA4eig, and AMCDE on selected real-world problems. These peers are chosen as BIBSDE's origin (IMODE), competition winners (NL-SHADE-RSP, EA4eig), and a recent journal publication (AMCDE). Table 16 presents results with Wilcoxon tests.

For five of eight problems (P1, P4, P10, P11.3, P11.5), all algorithms obtain solutions far better than the reference from L-SHADE-cnEpSin-PWI [35] for traditional search, confirming these problems merit long-term search. BIBSDE outperforms IMODE and NL-SHADE-RSP, shows no advantage over EA4eig, and is narrowly defeated by AMCDE. Overall, BIBSDE remains competitive on CEC 2011 problems, though advantages are less pronounced than on benchmark suites.

4.4. Observation on BIBSDE

BIBSDE improves IMODE through five behavioral revisions and one parameter change: (1) replacing LPSR with NLPSR, (2) reducing individuals just better than the worst, (3) linearly descending ϕ for mutation strategies, (4) uniformly setting and linearly descending CR in the second half, and (5) allocating best individuals to the best-performing strategy. Additionally, p in Equation 6 changes from 0.3 to 0.7.

An ablation study investigates these components (NLPSR (N), reduction scheme (R), ϕ descending (ϕ), CR adaptation (C), allocation scheme (A), and p_{new}) by comparing IMODE, IMODE+R, IMODE+C, IMODE+R+C+N, IMODE+R+C+pnew+N, IMODE+R+C+pnew+N+ϕ, and final BIBSDE on CEC 2020 at $D = 15$. Table 17 shows results, with Friedman test outcomes in Table 18 . Performance improves as more measures are added, confirming all proposals are beneficial.

4.5. Discussion

Experiments on CEC 2020/2022 show BIBSDE converges faster than IMODE and outperforms peers. CEC 2011 tests confirm competitiveness. The ablation study validates each measure's effectiveness, including the main contribution of rewarding the best-performing strategy and supporting modifications.

5. Conclusion

Long-term search in real-parameter single-objective optimization has been studied for several years, with low-convergence-velocity population-based metaheuristics proposed to address it. IMODE, a DE algorithm using three mutation strategies, shows strong performance among such metaheuristics. Based on IMODE, we propose BIBSDE, whose core idea rewards the best-performing mutation strategy with the best individuals. Five algorithmic revisions and one parameter adjustment are introduced. Experiments demonstrate BIBSDE's power for long-term search.

The success of BIBSDE reveals promising potential for reward mechanisms in ensemble construction, though reward details warrant further study—an avenue for future research.

References

[1] J. Brest, M. S. Maučec, B. Bošković, Differential evolution algorithm for single objective bound-constrained optimization: Algorithm j2020, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.

[2] G. Sun, G. Xu, N. Jiang, A simple differential evolution with time-varying strategy for continuous optimization, Soft Computing 24 (4) (2020) 2727–2747.

[3] J. Brest, M. S. Maučec, B. Bošković, Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: Algorithm j21, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 817–824.

[4] J. Cheng, Z. Pan, H. Liang, Z. Gao, J. Gao, Differential evolution algorithm with fitness and diversity ranking-based mutation operator, Swarm and Evolutionary Computation 61 (2021).

[5] V. Stanovov, S. Akhmedova, E. Semenkin, Nl-shade-rsp algorithm with adaptive archive and selective pressure for cec 2021 numerical optimization, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 809–816.

[6] G. Sun, C. Li, L. Deng, An adaptive regeneration framework based on search space adjustment for differential evolution, Neural Computing and Applications 33 (15) (2021) 9503–9519.

[7] Z. Tan, K. Li, Y. Wang, Differential evolution with adaptive mutation strategy based on fitness landscape analysis, Information Sciences 549 (2021) 142–

163.

[8] R. Biedrzycki, J. Arabas, E. Warchulski, A version of nl-shade-rsp algorithm with midpoint for cec 2022 single objective bound constrained problems, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1–8.

[9] V. Stanovov, S. Akhmedova, E. Semenkin, Nl-shade-lbc algorithm with linear parameter adaptation bias change for cec 2022 numerical optimization, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 01–08.

[10] M. Wang, Y. Ma, P. Wang, Parameter and strategy adaptive differential evolution algorithm based on accompanying evolution, Information Sciences 607 (2022) 1136–1157.

[11] G. Sun, B. Yang, Z. Yang, G. Xu, An adaptive differential evolution with combined strategy for global numerical optimization, Soft Computing 24 (9) (2020) 6277–6296.

[12] G. Yildizdan, Ö. K. Baykan, A novel modified bat algorithm hybridizing by differential evolution algorithm, Expert Systems with Applications 141 (2020) 112949.

[13] P. Civicioglu, E. Besdok, Bezier search differential evolution algorithm for numerical function optimization: A comparative study with crmlsp, mvo, wa, shade and lshade, Expert Systems with Applications 165 (2021) 113875.

[14] A. W. Mohamed, A. A. Hadi, P. Agrawal, K. M. Sallam, A. K. Mohamed, Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with imode algorithm for solving cec 2021 benchmark problems, in: 2021 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2021, pp. 841–848.

[15] P. Bujok, P. Kolenovsky, Eigen crossover in cooperative model of evolutionary algorithms applied to cec 2022 single objective numerical optimisation, in: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2022, pp. 1–8.

[16] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, M. J. Ryan, Improved multi-operator differential evolution algorithm for solving unconstrained problems, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.

[17] Z. Tan, K. Li, Differential evolution with mixed mutation strategy based on deep reinforcement learning, Applied Soft Computing 111 (2021) 107678.

[18] X. Wang, C. Li, J. Zhu, Q. Meng, L-shade-e: Ensemble of two differential evolution algorithms originating from l-shade, Information Sciences 552 (2021) 201–219.

[19] X. Xia, L. Gui, Y. Zhang, X. Xu, F. Yu, H. Wu, B. Wei, G. He, Y. Li, K. Li, A fitness-based adaptive differential evolution algorithm, Information Sciences 549 (2021) 116–141.

[20] C. Ye, C. Li, Y. Li, Y. Sun, W. Yang, M. Bai, X. Zhu, J. Hu, T. Chi, H. Zhu, et al., Differential evolution with alternation between steady monopoly and transient competition of mutation strategies, Swarm and Evolutionary Computation 83 (2023) 101403.

[21] X. Zhu, C. Ye, L. He, H. Zhu, T. Chi, J. Hu, Ensemble of differential evolution and gaining–sharing knowledge with exchange of individuals chosen based on fitness and lifetime, Soft Computing (2023) 1–16.

[22] J. Zhang, A. C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. on Evo. Comput. 13 (5) (2009) 945–958.

[23] A. W. Mohamed, A. A. Hadi, A. K. Mohamed, N. H. Awad, Evaluating the performance of adaptive gainingsharing knowledge based algorithm on cec 2020 benchmark problems, in: 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020, pp. 1–8.

[24] C. T. Yue, K. V. Price, J. J. Liang, M. Z. Ali, B. Y. Qu, N. H. Awad, P. P. Biswas, Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization, Tech. rep., School of Electrical Engineering, Zhengzhou University, Zhengzhou, China and 836 Owl Circle, Vacaville, CA, U.S.A. and School of EEE, Nanyang Technological University, Singapore and School of Computer Information Systems, Jordan University of Science and Technology, Jordan and School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou, China and Department of Computer Science, Faculty of Engineering, University of Freiburg and Advanced Digital Science Center, Singapore (2019).

[25] A. Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi, P. N. Suganthan, Problem definitions and evaluation criteria for the cec 2022 special session and competition on single objective bound constrained numerical optimization, Tech. rep., Department of Electrical Engineering, Indian Institute of Technology (BHU), Varanasi, Varanasi, India and Vacaville, California, USA and Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt and Department of Mathematics and Actuarial Science, School of Sciences Engineering, The American University in Cairo, Giza, Egypt and Faculty of Computer Studies, Arab Open University, Riyadh, Saudi Arabia and School of EEE, Nanyang Technological University, Singapore (2021).

[26] S. Das, P. N. Suganthan, Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems, Tech. rep., Dept. of Electronics and Telecommunication Engg., Jadavpur University, Kolkata, India and School of EEE, Nanyang Technological University, Singapore (2010).

[27] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, B. Y. Qu, Problem def-

initions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization, Tech. rep., School of EEE, Nanyang Technological University, Singapore and School of Computer Information Systems, Jordan University of Science and School of Electrical Engineering, Zhengzhou University, Zhengzhou, China (2016).

[28] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657.

[29] J. Brest, M. S. Maučec, B. Bošković, The 100-digit challenge: Algorithm jde100, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 19–26.

[30] V. Stanovov, S. Akhmedova, E. Semenkin, LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems, in: Proc of CEC, IEEE, 2018, pp.

[31] Y. Wang, H.-X. Li, T. Huang, L. Li, Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Applied Soft Computing 18 (2014) 232–247.

[32] P. Bujok, J. Tvrdík, Enhanced individual-dependent differential evolution with population size adaptation, in: 2017 IEEE congress on evolutionary computation (CEC), IEEE, 2017, pp.

[33] N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, in: Proceedings of IEEE international conference on evolutionary computation, IEEE, 1996, pp. 312–317.

[34] J. Brest, M. S. Maučec, B. Bošković, Single objective real-parameter optimization: Algorithm jSO, in: Proc. of CEC, IEEE, 2017, pp. 1311–1318.

[35] A. P. Piotrowski, L-shade optimization algorithms with population-wide inertia, Information Sciences 468 (2018) 117–141.

Note: Figure translations are in progress. See original paper for figures.