Complex & Intelligent Systems
https://doi.org/10.1007/s40747-023-01340-9

ORIGINAL ARTICLE

Emulation-based adaptive differential evolution: fast and auto-tunable approach for moderately expensive optimization problems

Kei Nishihara[1] · **Masaya Nakata**[1]

Received: 26 July 2023 / Accepted: 29 December 2023

Abstract
In the field of expensive optimization, numerous papers have proposed surrogate-assisted evolutionary algorithms (SAEAs) for a few thousand or even hundreds of function evaluations. However, in reality, low-cost simulations suffice for a lot of real-world problems, in which the number of function evaluations is moderately restricted, e.g., to several thousands. In such moderately restricted scenario, SAEAs become unnecessarily time-consuming and tend to struggle with premature convergence. In addition, tuning the SAEA parameters becomes impractical under the restricted budgets of function evaluations—in some cases, inadequate configuration may degrade performance instead. In this context, this paper presents a fast and auto-tunable evolutionary algorithm for solving moderately restricted expensive optimization problems. The presented algorithm is a variant of adaptive differential evolution (DE) algorithms, and is called emulation-based adaptive DE or EBADE. The primary aim of EBADE is to emulate the principle of sample-efficient optimization, such as that in SAEAs, by adaptively tuning the DE parameter configurations. Specifically, similar to Expected Improvement-based sampling, EBADE identifies parameter configurations that may produce expected-to-improve solutions, without using function evaluations. Further, EBADE incepts a multi-population mechanism and assigns a parameter configuration to each subpopulation to estimate the effectiveness of parameter configurations with multiple samples carefully. This subpopulation-based adaptation can help improve the selection accuracy of promising parameter configurations, even when using an expected-to-improve indicator with high uncertainty, by validating with respect to multiple samples. The experimental results demonstrate that EBADE outperforms modern adaptive DEs and is highly competitive compared to SAEAs with a much shorter runtime.

Keywords Adaptation · Differential evolution · Multi-population · Expensive optimization problem

Introduction

Several real-world applications, e.g., neural architecture search [64] and aerodynamic design [17], require optimization of expensive-to-evaluate objectives, where the objective values are calculated using computationally expensive simulations [44]. Considering vehicle structure optimization [39] as an example, the evaluation of a single design using a crashworthiness simulation takes 20 h [39]. For such expensive optimization problems (EOPs), the number of function evaluations (FEs) is restricted due to limited budgets of computational and financial resources. Consequently, the main challenge of EOPs is to obtain acceptable solutions under the restricted number of FEs. For this challenge, sample-efficient approaches, such as Bayesian optimization, that reduce the number of FEs are prevalent. Although EOPs are encountered for both single-objective and multi-objective optimization domains, this paper focuses on single-objective EOPs and denotes them as EOPs for simplicity.

Over the last two decades, various sample-efficient approaches have been developed based on evolutionary algorithms (EAs) [13]. The main motivation behind this pursuit is that typical EAs often assume hundreds of thousands of FEs and thereby become impractical in terms of solving EOPs. Several sample-efficient approaches consider very restricted FE budgets, usually with a few thousand or even hundreds of FEs. In addition to such extreme cases, there exist many

✉ Kei Nishihara
nishihara-kei-jv@ynu.jp

Masaya Nakata
nakata-masaya-tb@ynu.ac.jp

[1] Faculty of Engineering, Yokohama National University, 79-1, Tokiwadai, Hodogaya, Yokohama, Kanagawa 2408501, Japan

Published online: 15 February 2024

EOP instances using low-cost simulations, where budgets are moderately restricted to, e.g., several thousands of FEs. For example, in the automatic calibration of watershed models [54], a maximum of 10,000 FEs are used, and the corresponding evaluation of one solution using the Soil and Water Assessment Tool takes at least 2 min. However, such EOPs with moderately restricted budgets, referred to as moderately EOPs in this paper, have not undergone adequate systematic research compared to ones with very restricted budgets.

Surrogate-assisted EAs (SAEAs) [8, 16] are a popular sample-efficient approach for solving EOPs. Surrogate models of the objective function are constructed using machine learning (ML) techniques, and SAEAs utilize them to identify expected-to-improve solutions. For example, several works have proposed SAEAs using Bayesian optimization, where EAs are used to optimize the Expected Improvement (EI) metric [18]. SAEAs have been proven to be effective when the number of FEs is a few thousand or in the order of hundreds [13]. However, application of SAEAs to moderately EOPs in practice experiences the following difficulties.

- Other than their excellent performance corresponding to a few thousand of FEs, SAEAs tend to struggle with premature convergence when a higher number of FEs are considered [47, 60]. This problem has been highlighted more clearly in complex function landscapes [24].
- Owing to the aforementioned difficulty, tuning the parameter configurations of SAEAs, which govern their performance, is important [26, 32, 40]. However, advance fine-tuning under restricted FE budgets is usually hindered in EOPs.
- Most SAEAs are time-consuming as they repeatedly construct and reuse ML models during a run. Their runtimes are often not ascribed much importance, relying on an assumption that they are negligibly smaller than the computational times required by simulations [5, 10]. However, reducing the runtime becomes crucial when using low-cost simulations [5].

Note that the second difficulty is not only applicable to SAEAs but also EA-based algorithms; it is known that EA performances depend significantly on their parameter configurations [27]. Thus, despite the great success of SAEAs, development of high-performance, auto-tunable, and computationally efficient algorithms is essential.

Incorporating automatic parameter-tuning mechanisms into EAs is an effective approach to improve performance while avoiding manual parameter-tuning [20]. Adaptive EAs [19] are a popular paradigm in this regard—their parameter configurations are controlled during a run. For example, jDE [3] controls two parameters used in the differential evolution (DE) [46] algorithm—the scaling factor and the mutation rate. In addition to these two parameters, SaDE [42] controls mutation and crossover strategies of DE. Several works have proposed various adaptive EAs and established that they outperform standard EAs, with comparable runtimes [3, 23, 61].

Accordingly, adaptive EAs can be used to develop computationally efficient and auto-tunable methods for solving moderately EOPs. However, most adaptive EAs are not designed for this purpose as they usually assume hundreds of thousands of FEs [23, 41]. In particular, the adaptive EA, in its basic form, is subject to the following limitations when it is extended to an EOP. Most adaptive EAs are designed to update new parameter configurations based on ones that generate good solutions in past generations, and updated parameter settings are employed without validating their effectiveness in advance. This may require extensive trial-and-error to identify good parameter configurations and thereby requires a high number of FEs [29]. Further, most adaptive EAs are designed to assign different parameter configurations to each solution in a population, i.e., so-called individual-based adaptation. However, this adaptation style may be less effective in reliably identifying a good parameter configuration, because the effectiveness of each configuration is usually validated with respect to only one sample.

This paper presents a novel adaptive EA as a computationally efficient and auto-tunable approach for solving moderately EOPs. The presented algorithm, an emulation-based adaptive DE (EBADE), is based on the DE framework and does not utilize surrogate models. However, to improve sampling efficiency, EBADE is designed to emulate the principle of sample-efficient approaches, such as those in SAEAs, by controlling the parameter configurations. In particular, the following two strategies are involved in EBADE. First, a prior validation process is introduced to pre-screen candidate parameter configurations before use. In this process, as in EI-based sampling, candidates that likely generate "expected-to-improve" solutions are selected without using FEs. This intends to prevent the use of less-effective parameter configurations and thereby reduce the number of FEs. Second, EBADE employs a subpopulation-based adaptation, in which a parameter configuration is assigned to each subpopulation, rather than to each solution. Thus, each parameter configuration is used to produce multiple solutions to update its corresponding subpopulation. This intends to validate the effectiveness of parameter configurations using multiple samples carefully. That is, in EOP, parameter configurations should be evaluated by multiple samples. The main contributions of this work are as follows:

- To the best of our knowledge, this is the first attempt to develop an adaptive EA for moderately EOPs. This paper contributes to the development of computationally efficient and auto-tunable approaches for EOPs.

- An adaptation mechanism for parameter configurations effective under restricted FE budgets is introduced. Its effectiveness is validated by comparing EBADE with not only popular adaptive DEs but also state-of-the-art SAEAs.

Note that our previous work [37] presented an early investigation on the prior validation mechanism and integrated it into two popular adaptive DEs—jDE and SaDE. However, the prior validation process was designed to pre-screen candidate parameter configurations based on their ability to reproduce the current best solution, which suffers from premature convergence. Further, it was applicable only to individual-based adaptation frameworks like jDE and SaDE. Consequently, the prior validation-based jDE and SaDE underperform in the case of moderately EOPs compared to state-of-the-art SAEAs. EBADE extends the prior validation process to emulate EI-based sampling on the subpopulation-based adaptation framework.

The remainder of this paper is organized as follows. Section 2 describes the standard DE framework, as well as possible parameters and genetic operators considered as primary options in the present study. The latter half of Sect. 2 presents a literature review. Section 3 presents the detailed mechanism of EBADE. Section 4 reports experiments conducted using the CEC 2013 real-parameter single-objective benchmark function suite [25]. We compare the performances of EBADE and popular adaptive DEs as well as state-of-the-art SAEAs to investigate their effectiveness for moderately EOPs. Section 5 discusses the EBADE algorithm. Finally, Sect. 6 presents our conclusions and prospective directions of future research.

Background

This section describes the DE algorithm as background information. Subsequently, related works are summarized.

Differential Evolution

DE is a population-based evolutionary algorithm for solving a real-parameter optimization problem using a single-objective function, $f : \mathbb{R}^D \to \mathbb{R}$, where D denotes the problem dimension. In this paper, we consider the minimization of f, where the search space $\mathcal{X}$ is bounded by $\mathcal{X} \in [l_j, u_j]_{j=1}^D$.

During initialization, DE produces and then evaluates N initial solutions, forming an initial population $\mathcal{P} = \{\boldsymbol{x}_i\}_{i=1}^N$. Each initial solution is uniformly sampled from the search space $\mathcal{X}$. Subsequently, as the main loop, all solutions in $\mathcal{P}$ are updated via the following procedures. For each solution $\boldsymbol{x}_i$, a mutant solution $\boldsymbol{v}_i$ is produced using a defined mutation strategy with a scaling factor $F \in [0, 1]$. Note that the original paper considers $F \in [0, 2]$ [46], but $F \in [0, 1]$ is typically assumed [52]. Table 1 summarizes popular mutation strategies, where $\boldsymbol{x}_{r_1}$, $\boldsymbol{x}_{r_2}$, $\boldsymbol{x}_{r_3}$, $\boldsymbol{x}_{r_4}$, and $\boldsymbol{x}_{r_5}$ denote mutually exclusive solutions randomly selected from $\mathcal{P}$ and different from $\boldsymbol{x}_i$; $\boldsymbol{x}_{best}$ denotes the current best solution in $\mathcal{P}$; and $\boldsymbol{x}_{pbest}$ denotes a randomly selected solution from the top $\lfloor N \times p \rfloor$ solutions in $\mathcal{P}$ where $p \in [0, 1]$ is a hyperparameter to define greediness [63]. Next, a trial solution $\boldsymbol{u}_i$ is generated by applying a defined crossover strategy to $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ with a crossover rate $CR \in [0, 1]$. Typically, *binomial* or *exponential* crossover strategies are used, which are described in Algorithms 1 and 2, respectively, where $rand[0, 1]$ denotes a random value in $[0, 1]$ sampled from a unified distribution. After generating trial solutions for all the solutions, they are evaluated using f. If $\boldsymbol{u}_i$ is not worse than $\boldsymbol{x}_i$, i.e., $f(\boldsymbol{u}_i) \leq f(\boldsymbol{x}_i)$, $\boldsymbol{x}_i$ is updated using $\boldsymbol{u}_i$. These procedures are repeated until the termination criteria are satisfied.

Table 1 Popular mutation strategies for DE

Strategy	Definition
rand/1	$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$
rand/2	$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3}) + F(\boldsymbol{x}_{r_4} - \boldsymbol{x}_{r_5})$
best/1	$\boldsymbol{v}_i = \boldsymbol{x}_{best} + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
best/2	$\boldsymbol{v}_i = \boldsymbol{x}_{best} + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}) + F(\boldsymbol{x}_{r_3} - \boldsymbol{x}_{r_4})$
current-to-rand/1	$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$
current-to-best/1	$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{best} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
current-to-pbest/1	$\boldsymbol{v}_i = \boldsymbol{x}_i + F(\boldsymbol{x}_{pbest} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
rand-to-best/1	$\boldsymbol{v}_i = \boldsymbol{x}_{r_1} + F(\boldsymbol{x}_{best} - \boldsymbol{x}_{r_1}) + F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$

Algorithm 1 Binomial crossover

Require: a base solution $\boldsymbol{x}$, a mutant solution $\boldsymbol{v}$, a crossover rate CR
Set $\boldsymbol{u}$ as a duplicate of $\boldsymbol{x}$
Set j_{rand} to a random integer sampled from $\{1, 2, \ldots, D\}$
for $j = 1$ **to** D **do**
  if $rand[0, 1] \leq CR$ **or** $j = j_{rand}$ **then**
    Set j-th variable of $\boldsymbol{u}$ to that of $\boldsymbol{v}$ as $u_j = v_j$
  end if
end for
return $\boldsymbol{u}$

Algorithm 2 Exponential crossover

Require: a base solution $\boldsymbol{x}$, a mutant solution $\boldsymbol{v}$, a crossover rate CR
Set $\boldsymbol{u}$ as a duplicate of $\boldsymbol{x}$
Set j to a random integer sampled from $\{1, 2, \ldots, D\}$
Set k as $k = 1$
repeat
  Set jth variable of $\boldsymbol{u}$ to that of $\boldsymbol{v}$ as $u_j = v_j$
  Update j as $j = (j + 1) \bmod D$
  Update k as $k = k + 1$
until $rand[0, 1] \geq CR$ **or** $k \geq D$
return $\boldsymbol{u}$

Table 2 List of adaptive and surrogate-assisted DEs for single-objective optimization

Algorithm	Adaptation style	D	$FE_{\max}$
Adaptive DEs			
jDE [3]	Indiv	{2, 4, 30}	10,000–20,000,000
FDSADE [53]	Indiv	{2, 4, 30}	50,000
ISADE [15]	Indiv	30	300,000
JADE [63]	Indiv	{2, 3, 4, 6, 30, 100}	6000–8,000,000
MDE_pBX [14]	Indiv	{30, 50, 100}	$D\times$10,000
SHADE [50]	Indiv	30	300,000
L-SHADE [51]	Indiv	{10, 30, 50, 100}	$D\times$10,000
jSO [4]	Indiv	{10, 30, 50, 100}	$D\times$10,000
SaDE [42]	Indiv	{10, 30}	100,000–500,000
CoDE [55]	Indiv	30	300,000
EPSDE [35]	Indiv	{10, 30, 50}	$D\times$10,000
CSDE [48]	Indiv	{30, 50, 100}	$D\times$10,000
AL-SHADE [23]	Indiv	{10, 30, 50}	$D\times$10,000
DE-DDQN [45]	Indiv	{10, 30}	10,000
FLDE [49]	Indiv	{10, 30, 50, 100}	$D\times$10,000
DE with two subpopulations [31]	Subpop	30	300,000
MPEDE [58]	Subpop	{30, 50}	$D\times$10,000
HMJCDE [22]	Subpop	{30, 50}	$D\times$10,000
EDEV [59]	Subpop	{30, 50}	$D\times$10,000
Surrogate-assisted DEs			
CADE [30]	–	30	{10,000, 20,000}
CRADE [28]	–	{30, 500}	10,000
GPEME [26]	–	{20, 30, 50}	1000
ESAO [57]	–	{20, 30, 50, 100, 200}	1000
SAHO [40]	–	{10, 20, 30, 50, 100}	{110, 220, 330, 1000}
DSS-DE [32]	–	{30, 50, 100}	1000
SADE-ATDSC [38]	–	{10, 30, 50, 100}	1000
DE-AEC [62]	Indiv	{2, 3, 4, 6}	100,000
S-JADE [7]	Indiv	{20, 30, 50, 100, 200}	{1000, 1500, 2000}
SMDE [21]	Indiv	{10, 25, 60, 72, 942}	12,000
DESSA [29]	Indiv	30	3000
SMA-EPSDE [33]	Indiv	{10, 30}	$D\times$10,000
ESMDE [34]	Indiv	{10, 30}	$D\times$10,000
Sa-DE-DPS [11]	Indiv	{10, 20, 30}	$D\times$ 50
SAPDE-ANN, SAPDE-RSM [1]	Indiv	{10, 30}	$D\times$10,000
EBADE (Proposed algorithm)	Subpop	{10, 20, 30}	6000

Columns "D" and "$FE_{\max}$" list the problem dimension and the maximum number of fitness evaluations adopted in the experiments, respectively

Related works

This subsection first reviews popular adaptive DEs. Then, surrogate-assisted DEs are introduced. Finally, the position of EBADE in this context is discussed.

Table 2 summarizes the related works discussed below. In the "Adaptation style" column, related works are categorized into two classes—Indiv. and Subpop. corresponding to individual-based and subpopulation-based adaptation, respectively. For algorithms without adaptation of any parameter configuration, the entry in this column is set to "–". The problem dimension and maximum number of FEs adopted in the experiments are listed in the columns "D" and "$FE_{\max}$", respectively, to indicate the type of problem addressed.

Adaptive DEs

Individual-based adaptation

In the jDE [3] framework, each solution is paired with specific values of F and CR, i.e., individual-based adaptation. This may potentially provide suitable parameter configurations for a particular solution. The hyperparameters F and CR are randomly sampled once again with predefined probabilities τ_F and τ_{CR}. They can then be updated based on comparisons between a solution and a trial solution in terms of fitness. This comparison-based adaptation utilizes the algorithm characteristic of the DE framework [52]. Several branches of jDE have been proposed, including FDSADE [53] and ISADE [15], which adaptively control both τ_F and τ_{CR}. JADE [63] explored another paradigm of the sampling method of the hyperparameters utilizing probabilistic distributions determined using previous information concerning superior solutions, which is different from uniform random sampling used in jDE. JADE revealed the impact of the sampling method on adaptive DEs, and several subsequent variants have incorporated the concept of JADE, e.g., MDE_pBX [14] and SHADE [50]. SHADE is a highly popular variant among adaptive DEs that updates probabilistic distributions using success-history memories. This mechanism improves the robustness of JADE by maintaining a diverse set of parameters of probabilistic distributions. Modern approaches, including L-SHADE [51] and jSO [4], utilize the SHADE framework to control F and CR. They also control N using linear population size reduction, which promotes a transition from exploration to exploitation with the progression of the search phase. Additionally, jSO employs fine-tuned scheduling-based adaptation.

Some adaptive DEs control both genetic operators and hyperparameters as parameter configurations to improve the capacity to specialize the framework to a given problem. However, this suffers from the drawback of increased complexity due to the increase in the number of parameter configurations requiring adaptation. One possible approach to address the bottleneck involves predefined sets of parameter configurations, e.g., SaDE [42] defines a limited number of pairs of mutations and crossovers, e.g., *rand/1/bin* and *rand-to-best/2/bin*. Similarly, CoDE [55] prepares predefined parameter sets $\{F, CR\}$ as well as pairs of genetic operators. EPSDE [35] adapts parameter configurations selected from two pools of hyperparameters and genetic operators. Recently, the selection of the best mutation strategy in adaptive DEs from all candidates has been studied. CSDE [48] and AL-SHADE [23] are state-of-the-art adaptive DEs that demonstrate that adaptation of mutation strategies remains important while proposing new mutation strategies. DE-DDQN [45] and FLDE [49] primarily adapt to mutation strategies via reinforcement learning and random forest, respectively, but these ML-based methods require long computational times.

Subpopulation-based adaptation

Recent studies have integrated a multi-population scheme into the DE framework to utilize the divide-and-conquer strategy. They divide the population into multiple subpopulations, with each searching a different area. When the adaptation mechanism is included in a multi-population scheme, different parameter configurations of DE are assigned to subpopulations. For example, two populations are used for exploitation and exploration in the study [31]. In MPEDE [58], three defined parameter configurations are paired with corresponding subpopulations, where each subpopulation size is adaptively controlled depending on the current search dynamics. Moreover, some works allocate different adaptation mechanisms of existing adaptive DEs to subpopulations to further improve performance by combining various adaptation mechanisms. For instance, a hybrid amalgamation of CoDE and JADE was proposed as HMJCDE [22], and EDEV [59] incorporates EPSDE, CoDE, and JADE.

Surrogate-assisted DEs

Surrogate-assisted DEs without adaptation of DE parameter configurations

In EOPs, especially under severely restricted FE budgets, SAEAs comprise one of the most popular approaches. We now introduce some surrogate-assisted DEs. CADE [30] uses a support vector machine (SVM) as the classification model to screen solutions, reducing the number of FEs. CRADE [28] combines two SVMs to approximate the objective function and classify solutions, compensating for the weaknesses of both models. GPEME [26] is a standard SAEA, which reduces dimension using the Sammon mapping when the problem dimension exceeds 30. Then, it generates offspring solutions and evaluates only one solution that is "expected-to-improve" using the lower confidence bound metric obtained by the Kriging model constructed using recently evaluated solutions. ESAO [57] combines the global radial basis function network (RBFN) and the local Kriging model. RBFN is used to roughly select a search region and the elaborate search is conducted using the Kriging model. SAHO [40] adaptively selects optimizers from DE and TLBO [43] while screening solutions using RBFN to design diverse searches. Recently, SAEAs that adaptively select RBFNs with different configurations have been proposed, e.g., DSS-DE [32] and SADE-ATDSC [38], since the performances of SAEAs depend on parameter configurations of both EAs and MLs.

Surrogate-assisted DEs with individual-based adaptation

Some surrogate-assisted DEs incorporate adaptive DE mechanisms into SAEAs to inherit the advantages of both EAs and SAEAs. However, most of these algorithms use

individual-based adaptation—SAEAs with subpopulation-based adaptation have not been developed. For example, DE-AEC [62] uses RBFN to screen solutions and adapts F and CR, as in jDE. S-JADE [7] enhances the performance of JADE using RBFN. Like ESAO, S-JADE [7] utilizes two types of RBFNs; global and local RBFNs are constructed using all evaluated solutions and neighborhood solutions of the current population, respectively.

Alternatively, DE parameter configurations have also been adapted while screening solutions. This produces various solutions to be screened by varying DE parameter configurations. SMDE [21] uses four mutation strategies and generates the same number of offspring solutions via mutation strategies for each base solution. Then, the solution with the best predicted fitness is adopted. Thus, SBSM-DE considers four candidate solutions generated in different ways, although FE is conducted only once per base solution. Other examples are as follows. DESSA [29] is used with CoDE or SaDE. For example, DESSA-CoDE has three sets of parameters and three mutation strategies, i.e., nine pairs of parameter configurations are considered as adaptation candidates. After generating the nine corresponding trial solutions, rank-SVM is used as a surrogate model for screening. SMA-EPSDE [33] and ESMDE [34] are derivatives of EPSDE and use two mutation strategies (*rand/1* and *current-to-rand/1*), two crossover strategies (*binomial* and *exponential*), $F \in [0.5, 1.0]$, and $CR \in [0.0, 1.0]$ as candidate parameter configurations. They randomly sample parameter configurations and generate solutions until the approximated fitness exceeds the fitness of base solutions. The difference between SMA-EPSDE and ESMDE comprises the dataset selection criteria for the Kriging model. Sa-DE-DPS [11], SAPDE-ANN [1], and SAPDE-RSM [1] conduct a search on the approximation function for a certain number of generations while adapting parameter configurations to accelerate the search.

Position of EBADE in this context

According to Table 2, most adaptive DEs are tested with $D\times$ 10,000 FEs, while many surrogate-assisted DEs are designed for FEs not exceeding 1000. Some surrogate-assisted DEs which screen solutions and adapt parameter configurations simultaneously use $D\times$ 10,000 FEs, but real-world EOPs with more than 10,000 FEs are rarely encountered [2, 9, 12, 17, 21, 32, 36, 56, 57]. Thus, we investigate if EBADE improves performance within 10,000 FEs, i.e., for moderately EOPs.

As noted in the "Adaptation style" column in Table 2, most existing adaptive DEs use individual-based adaptation, while subpopulation-based adaptation is only being studied recently. In this study, EBADE adapts parameter configurations using subpopulation-based adaptation, without using surrogate models. In the experiments presented in Sect. 4, we compare EBADE with adaptive DEs and SAEAs. For adaptive DEs, we choose comparison algorithms based on both individual-based and subpopulation-based adaptation to investigate the effectiveness of EBADE compared to both typical and modern categories.

The prior validation mechanism of EBADE introduced in the next section has never been discussed in the existing literature. In contrast, most adaptive EAs update parameter configurations by referring to those that have generated good solutions in past generations.

Proposed algorithm

The proposed algorithm, EBADE, is an adaptive DE for solving EOPs. In contrast to typical adaptive EAs, a prior validation process is utilized to identify good DE parameter configurations before using them. This accelerates the evolutionary search while reducing the number of FEs by avoiding the use of less effective parameter configurations. Here, we first discuss the idea underlying our prior validation process, i.e., the process of estimating the effectiveness of parameter configurations in advance, and subpopulation-based adaptation. Subsequently, the detailed algorithms of EBADE are explained.

Concept

Usually, the effectiveness of algorithmic parameter configurations cannot be estimated without using them, because it depends on the current search dynamics and random factors. However, the idea of EBADE is to produce good parameter configurations by emulating the sampling mechanism of SAEA without consuming FEs. Specifically, EBADE comprises the prior validation mechanism and subpopulation-based adaptation. Our idea is based on two insights summarized below.

- Generally, SAEAs first generate multiple candidate solutions and estimate the quality of candidate solutions, and then remove undesired ones to reduce FEs. When estimating quality, SAEAs often utilize the improvability of candidate solutions, e.g., the EI metric. However, EBADE cannot estimate the quality of candidate solutions directly due to the lack of surrogates. Instead, EBADE utilizes parameter configurations that directly affect solution generation. By emulating the SAEA mechanism, EBADE generates multiple parameter configurations and estimates their quality. Then, without consuming FEs, EBADE removes less effective parameter configurations before generating solutions for FE. Emulating the idea of the EI metric, EBADE selects a

candidate parameter configuration that is likely to generate "expected-to-improve" solutions. Specifically, the superior solution mentioned in the preceding paragraph is set to the solution with the best fitness improvement ratio (FIR). Since FIR represents the degree of improvement in the fitness value, the solution with the best FIR may not necessarily coincide with the solution with the best fitness value, preventing the latter from being chosen even if it is stuck in a local optimum. Also, by selecting the solution with the best FIR, the various solutions that are progressing will continue to be selected even if some solutions in the population stagnate. Thus, our prior validation mechanism is expected to enable EBADE to guide solutions to the "expected-to-improve" area, transcending premature convergence to the local optima.

- Although testing or validation data consist of multiple samples in the ML domain, most existing adaptive DEs utilize individual-based adaptation. In other words, a parameter configuration is validated using only one solution in the existing adaptive DEs. Thus, individual-based adaptation may be inefficient in EOPs, because a large number of FEs are consumed over multiple generations to gain a large number of samples for a parameter configuration during its validation. Accordingly, EBADE employs subpopulation-based adaptation, where a parameter configuration is paired with multiple solutions, i.e., all solutions in a subpopulation. This improves the validation accuracy of the effectiveness of parameter configurations, reflecting the results of FE by narrowing down the number of parameter configurations and validating each with respect to multiple samples. Additionally, subpopulation-based adaptation is also vital for the prior validation mechanism as one candidate parameter configuration is validated with respect to multiple samples in the prior validation phase. The subpopulation-based scheme can mitigate the possibility of a sample moving in an unintended direction due to random numbers, resulting in an unjustified evaluation of parameter configurations.

Consequently, EBADE is designed to improve the efficiency of adaptation using the prior validation mechanism and subpopulation-based adaptation, emulating sample-efficient approaches such as SAEAs without using any surrogate.

Parameter configuration vector

In EBADE, four algorithmic parameter configurations, the mutation strategy, the crossover strategy, the scaling factor, and the crossover rate are controlled during a run. Let $\boldsymbol{\theta} = [\theta_v, \theta_u, \theta_F, \theta_{CR}]$ be a *parameter configuration vector* that defines the DE parameter configuration, where each variable in $\boldsymbol{\theta}$ is defined as follows:

Algorithm 3 Get trial solution

Require: a parameter configuration vector $\boldsymbol{\theta}$, a base solution $\boldsymbol{x}$, a population $\mathcal{P}$

/** Generate a mutant solution $\boldsymbol{v}$ **/
switch θ_v **do**
  case 1
    $\boldsymbol{v} = \boldsymbol{x}_{best} + \theta_F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
  case 2
    $\boldsymbol{v} = \boldsymbol{x} + \theta_F(\boldsymbol{x}_{best} - \boldsymbol{x}) + \theta_F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
  case 3
    $\boldsymbol{v} = \boldsymbol{x} + \theta_F(\boldsymbol{x}_{pbest} - \boldsymbol{x}) + \theta_F(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2})$
  case 4
    $\boldsymbol{v} = \boldsymbol{x}_{r_1} + \theta_F(\boldsymbol{x}_{best} - \boldsymbol{x}_{r_1}) + \theta_F(\boldsymbol{x}_{r_2} - \boldsymbol{x}_{r_3})$
  where $\boldsymbol{x}_{r_{1,2,3}}$, $\boldsymbol{x}_{best}$, and $\boldsymbol{x}_{pbest}$ are selected from $\mathcal{P}$
/** Generate a trial solution $\boldsymbol{u}$ **/
switch θ_u **do**
  case 1
    $\boldsymbol{u} = BinomialCrossover(\boldsymbol{x}, \boldsymbol{v}, \theta_{CR})$ — see Algorithm 1
  case 2
    $\boldsymbol{u} = ExponentialCrossover(\boldsymbol{x}, \boldsymbol{v}, \theta_{CR})$ — see Algorithm 2
return $\boldsymbol{u}$

- $\theta_v \in \{1, 2, 3, 4\}$ specifies the index of the mutation strategy to be used. EBADE uses four mutation strategies, *best/1*, *current-to-best/1*, *current-to-pbest/1*, and *rand-to-best/1*, indexed by 1, 2, 3, and 4, respectively.
- $\theta_u \in \{1, 2\}$ specifies the index of the crossover strategy to be used. *Binomial* and *exponential* crossover strategies are used and indexed by 1 and 2, respectively.
- $\theta_F \in [0, 1]$ indicates the specific value of the scaling factor F used in the mutation strategy specified by θ_v.
- $\theta_{CR} \in [0, 1]$ indicates the specific value of the crossover rate CR used in the crossover strategy specified by θ_u.

For example, given $\boldsymbol{\theta} = [1, 2, 0.5, 0.8]$, a DE algorithm uses *best/1* mutation with $F = 0.5$ and *exponential* crossover with $CR = 0.8$. As the mutation strategy, we choose those that accelerate convergence of the DE population the most [6], since the rapid convergence of the population is prioritized during the optimization of EOPs [16].

In summary, the solution-generation procedure of DE specified with $\boldsymbol{\theta}$, i.e., the generation of trial solutions, is described in Algorithm 3.

Overall framework

EBADE uses and subsequently optimizes M subpopulations, $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$, simultaneously, where each subpopulation size is set to a common value N. Each subpopulation $\mathcal{P}_m$ has its own parameter configuration vector $\boldsymbol{\theta}_m$.

Algorithm 4 describes the overall procedure of EBADE, which consists of the following four components—initialization, a search phase, post hoc validation, and prior validation. To begin with, each subpopulation $\mathcal{P}_m$ and its parameter configuration vector $\boldsymbol{\theta}_m$ are initialized. Next, in the search phase, DE is executed for each subpopulation with its corresponding parameter configuration. Subsequently, post hoc validation is performed to identify ineffective parameter configurations that have failed to produce good solutions during the search phase. Finally, ineffective parameter configurations are modified in the prior validation phase into plausibly good parameter configurations without consuming FEs. Subsequently, EBADE returns to the search phase with the modified parameter configurations and the three latter phases are repeated until the termination criteria are satisfied.

The rest of this section describes the detailed procedure of each phase.

Algorithm 4 EBADE

Require: problem dimension D, search space $\mathcal{X} \in [l_j, u_j]_{j=1}^{D}$, subpopulation size N, the number of subpopulations M, the number of candidate parameter configurations K

1: /** Initialization **/
2: Initialize M subpopulations, $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$, where each subpopulation is composed of N initial solutions randomly generated in $\mathcal{X}$
3: Evaluate all initial solutions
4: Initialize M parameter configuration vectors $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M$
5: **while** *termination criteria are not satisfied* **do**
6: /** Search phase **/
7: Construct the whole population $\mathcal{P}_{all} = \bigcup_{m=1}^{M} \mathcal{P}_m$
8: **for** $m = 1$ **to** M **do**
9: **for each** $\boldsymbol{x} \in \mathcal{P}_m$ **do**
10: $\boldsymbol{u} \leftarrow GetTrialSolution(\boldsymbol{\theta}_m, \boldsymbol{x}, \mathcal{P}_{all})$ — see Algorithm 3
11: Evaluate $\boldsymbol{u}$
12: **if** $f(\boldsymbol{u}) \leq f(\boldsymbol{x})$ **then**
13: $\boldsymbol{x} = \boldsymbol{u}$
14: **end if**
15: **end for**
16: **end for**
17: /** Post hoc validation phase **/
18: Construct the whole population $\mathcal{P}_{all} = \bigcup_{m=1}^{M} \mathcal{P}_m$
19: Calculate FIR of $\forall \boldsymbol{x} \in \mathcal{P}_{all}$
20: Select top M solutions $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_M^*$ from $\mathcal{P}_{all}$ in terms of FIR
21: $\Theta = \{i \in \{1, 2, \ldots, M\} \mid \nexists \boldsymbol{x}_m^* \in \mathcal{P}_i, m = 1, 2, \ldots, M\}$
22: /** Prior validation phase **/
23: **for each** $i \in \Theta$ **do**
24: $\boldsymbol{\theta}_i$ is modified by executing *PriorValidation* using K — see Algorithm 5
25: **end for**
26: **end while**

Initialization

Initially, each subpopulation $\mathcal{P}_m$ is initialized with N initial solutions. The initial solutions are produced using the same method as the standard DE (see Sect. 2.1). Accordingly, similar to existing DEs with multi-population mechanisms [22, 31, 58, 59], EBADE produces different initial subpopulations to boost the performance while preventing premature convergence. Thereafter, all initial solutions are evaluated using the objective function. Subsequently, the parameter configuration vector $\boldsymbol{\theta}_m$ for $\mathcal{P}_m$ is initialized as follows. The values of θ_v and θ_u are set to random integer values sampled from $\{1, 2, 3, 4\}$ and $\{1, 2\}$, respectively. For θ_F and θ_{CR}, $\theta_F = 0.5$ and $\theta_{CR} = 0.9$ are used as the default configurations of standard DE [46].

Search phase

This phase conducts evolutionary search to collect multiple validation samples and estimate the effectiveness of parameter configurations during the following post hoc validation phase. To this end, each subpopulation is updated by executing the DE solution-generation process for one generation.

Lines 6-16 in Algorithm 4 describe the procedure of this phase. A DE algorithm is executed on each subpopulation for one generation, corresponding to the update of all N solutions in $\mathcal{P}_m$. First, EBADE constructs the whole population $\mathcal{P}_{all}$ by concatenating all subpopulations, $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$. This population $\mathcal{P}_{all}$ is used as the pool of candidate solutions in the mutation strategy. Since this process also influences the discovery of superior solutions, we employ an *information-sharing* strategy that utilizes all possible solutions. In other words, all possible variables in $\mathcal{P}_{all}$, i.e., $\boldsymbol{x}_{r_{1,2,3}}$, $\boldsymbol{x}_{best}$, and $\boldsymbol{x}_{pbest}$, are used while generating a mutant solution $\boldsymbol{v}$ for $\boldsymbol{x} \in \mathcal{P}_m$, even if they belong to other subpopulations.

Next, all solutions in all subpopulations are updated following the original DE methodology. Specifically, for each subpopulation $\mathcal{P}_m$, each trial solution $\boldsymbol{u}$ of $\boldsymbol{x}$ is generated using $\boldsymbol{\theta}_m$ and $\mathcal{P}_{all}$. Then, each $\boldsymbol{u}$ is evaluated using the objective function. If the fitness value $f(\boldsymbol{u})$ of $\boldsymbol{u}$ does not exceed that of $\boldsymbol{x}$, $\boldsymbol{x}$ is replaced with $\boldsymbol{u}$. During the search phase, $N \times M$ solutions can be sampled for each $\boldsymbol{\theta}$; thus, exactly $N \times M$ FEs are consumed.

Post hoc validation phase

This phase identifies good/bad parameter configurations by estimating their effectiveness based on the results of the

previous search phase. *Good* parameter configurations are reused during the next search phase without any modification, whereas *bad* ones are modified during the next prior validation phase.

A straightforward way to define a good parameter configuration is in terms of its success to generate the current best solution in terms of the objective values. Although this definition was used in our previous work [37], it may easily induce premature convergence, as it does not consider the improvement of objective values. Here, we define good parameter configurations as "worth-to-continue" ones, which are identified as those generating solutions with high FIR values. Specifically, suppose a solution $\boldsymbol{x}_g$ is generated during the gth generation from its corresponding parent solution $\boldsymbol{x}_{g-1}$, and the FIR value for $\boldsymbol{x}_g$, denoted as $\delta_f(\boldsymbol{x}_g)$, is calculated as

$$\delta_f(\boldsymbol{x}_g) = 1 - \frac{f(\boldsymbol{x}_g)}{f(\boldsymbol{x}_{g-1}) + \delta_C}, \tag{1}$$

where $\delta_C \geq 0$ denotes a constant value to avoid division by zero. Corresponding to a large value of $\delta_f(\boldsymbol{x}_g)$, it is expected that the parameter configuration vector $\boldsymbol{\theta}$ used to generate $\boldsymbol{x}_g$ from $\boldsymbol{x}_{g-1}$ can be effective to identify further good solutions during the next search phase. In contrast, corresponding to a small value of $\delta_f(\boldsymbol{x}_g)$, $\boldsymbol{\theta}$ may be ineffective as it does not contribute to discovering a good search region.

Accordingly, EBADE determines good/bad parameter configurations using the following procedures. They are described by the lines 17-21 in Algorithm 4. First, EBADE reconstructs the whole population $\mathcal{P}_{all}$ by combining all subpopulations updated during the search phase. Subsequently, an FIR value for each solution $\boldsymbol{x}$ in $\mathcal{P}_{all}$ is calculated, and the top M solutions having the M highest FIR values, $\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \ldots, \boldsymbol{x}_M^*$, are determined. Then, a parameter configuration vector $\boldsymbol{\theta}_m$ is identified as a good one if there is at least one top solution $\boldsymbol{x}^*$ generated by $\boldsymbol{\theta}_m$; otherwise, it is identified as a bad one. Technically, EBADE stores indices of the bad parameter configurations in an index set of ineffective parameter configurations Θ, which is mathematically defined as follows:

$$\Theta = \{i \in \{1, 2, \ldots, M\} \mid \nexists \boldsymbol{x}_m^* \in \mathcal{P}_i, m = 1, 2, \ldots, M\}. \tag{2}$$

Note that EBADE utilizes FIR to determine good parameter configurations, but this may be hindered until solutions begin to improve to some extent after several generations. Other possible indicators may be required to detect good configurations even with a small improvement, but we will leave this for future work.

Prior validation phase

This process modifies bad parameter configurations to likely good ones before using them in the next search phase. As defined during post hoc validation, we consider good parameter configurations to be those that have generating good solutions with the top M FIR values. Using this definition, we modify bad parameter configurations to enable them to generate such top solutions.

Lines 22-25 in Algorithm 4 correspond to this phase and Algorithm 5 presents the detailed procedure of our prior validation phase. First, EBADE selects the target solution $\boldsymbol{x}_{target}$ from the solutions discovered so far. This is commonly set for all bad parameter configurations $\boldsymbol{\theta}_i$ to improve the probability of generating effective parameter configurations using $\boldsymbol{x}_{target}$ repeatedly. The solution with the best FIR in $\mathcal{P}_{all}$ is selected as the target solution. Note that the search direction is also guided towards the best solution as mutation strategies with high exploitation ability are utilized. Thus, the modified parameter configuration is expected to consider the directions towards the best solution and the area that remains to be searched, i.e., the best FIR solution.

For each bad parameter configuration indexed by $i \in \Theta$, EBADE randomly generates multiple parameter configuration candidates and then removes less effective ones, only retaining one. Then, the remaining candidate is replaced with $\boldsymbol{\theta}_i$. Technically, K candidate parameter configurations $\boldsymbol{\theta}_k$, $k = 1, 2, \ldots, K$ are generated following the procedure used during initialization, except for θ_F and θ_{CR}, where $K \in \mathbb{N}$ is a hyperparameter. For each $\boldsymbol{\theta}_k$, its θ_F and θ_{CR} are randomly selected from $[0, 1]$. Then, EBADE selects one candidate parameter configuration that produces a solution closest to a target solution $\boldsymbol{x}_{target}$. In particular, for each candidate parameter configuration $\boldsymbol{\theta}_k$, EBADE tests the ability of $\boldsymbol{\theta}_k$ to generate solutions close to a target solution $\boldsymbol{x}_{target}$. More specifically, for each solution $\boldsymbol{x}$ in its corresponding subpopulation $\mathcal{P}_i$, its trial solution $\boldsymbol{u}$ is generated by the DE solution-generation process specified using $\boldsymbol{\theta}_k$, i.e., $GetTrialSolution(\boldsymbol{\theta}_k, \boldsymbol{x}, \mathcal{P}_{all})$. These sample solutions are not evaluated using the objective function. Then, the minimum Euclidean distance between each $\boldsymbol{u}$ and $\boldsymbol{x}_{target}$ is recorded as $d(k)$. Finally, the index having the minimum $d(k)$, $k = 1, 2, \ldots, K$, is set as k^*, and $\boldsymbol{\theta}_i$ is replaced with the candidate $\boldsymbol{\theta}_{k^*}$.

Experiments

In this section, the performance of EBADE is evaluated on single-objective benchmark functions with restricted FE budgets. All experiments are conducted using a Intel(R) Core(TM) i7-10700 4.8 GHz CPU and 16 GB RAM.

Algorithm 5 Prior validation

Require: subpopulation size N, the number of candidate parameter configurations K

1: $\boldsymbol{x}_{target} = \arg\max_{\boldsymbol{x} \in \mathcal{P}_{all}} \delta_f(\boldsymbol{x})$
2: Sample K random candidate parameter configurations $\boldsymbol{\theta}_k$
3: **for** $k = 1$ **to** K **do**
4: Generate N trial solutions $\boldsymbol{u}$ from $\mathcal{P}_i$ by *GetTrialSolution*$(\boldsymbol{\theta}_k, \boldsymbol{x}, \mathcal{P}_{all})$
5: $d(k) \leftarrow$ the minimum distance between each pair of $\boldsymbol{u}$ and $\boldsymbol{x}_{target}$
6: **end for**
7: $k^* = \arg\min_{k \in \{1,2,\ldots,K\}} d(k)$
8: **return** $\boldsymbol{\theta}_{k^*}$

Experimental configurations

Test problems

We used 28 bound-constrained benchmark functions, F1, F2, ..., and F28, used in the competition on Real-Parameter Single Objective Optimization at the IEEE Congress on Evolutionary Computation 2013 [25]. Note that F1–F5, F6–F20, and F21–28 are categorized as unimodal, multimodal, and composition functions, respectively (see [25] for detailed definitions). The problem dimension was set to $D \in \{10, 20, 30\}$. The search space of all functions was commonly set to $\boldsymbol{x} \in [-100, 100]^D$.

Comparison algorithms

We compared the performances of EBADE and four adaptive DEs, SHADE, jSO, CSDE, and EDEV, and four SAEAs, GPEME, S-JADE, SAHO, and ESMDE. The brief descriptions and parameter configurations of comparison algorithms are summarized below.

- SHADE, which has been the basis of modern adaptive DEs, employs the *current-to-pbest/1* mutation strategy and adapts F and CR based on success history memories, $M_{F,r}$ and $M_{CR,r}$, respectively, using individual-based adaptation. Via comparison with SHADE, the impacts of our prior validation and subpopulation-based adaptation mechanisms are evaluated. We used $H = 100$, $M_{F,h,init} = 0.5$, $M_{CR,h,init} = 0.5$, $|Archive| = 100$, $p_{\min} = 2/N$, and $p_{\max} = 0.2$ [50].
- jSO, which is an extension of L-SHADE [51]. Based on a specific schedule for parameter control, jSO adapts F, CR, and N for individual-based adaptation. In particular, N decreases as the number of FEs increases to evolve more generations by the end of the search. We verify if EBADE outperforms local search conducted by jSO at the end of the search in moderately EOPs. We used $H = 5$, $M_{F,h,init} = 0.3$, $M_{CR,h,init} = 0.8$, $|Archive| = N$, $N_{init} = 25 \log D^{3/2}$, $N_{\min} = 4$, $p_{\min} = 0.125$, and $p_{\max} = 0.25$ [4].
- CSDE, which is one of the state-of-the-art adaptive DEs. CSDE adapts F, CR, and the mutation strategy for individual-based adaptation. CSDE shifts between two mutation strategies (*current-to-pbest/1* and *pbest-to-rand/1*) depending on the degree of stagnation in the search. Comparison with CSDE reveals if EBADE outperforms state-of-the-art adaptive DEs in moderately EOPs. We used $F_{init} = 0.5$, $CR_{init} = 0.5$, $N = 100$, $FP = 200$, $\mu = 0.5$, and $\sigma = 0.1$ [48].
- EDEV, which is a state-of-the-art subpopulation-based adaptive DE. EDEV adaptively assigns JADE, CoDE, and EPSDE to subpopulations. Thus, EDEV adapts F, CR, the mutation strategy, and the crossover strategy. Comparison with EDEV is important, because both EDEV and EBADE are subpopulation-based adaptive DEs that adapt four parameter configurations of DE. We used $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$, $\lambda_4 = 0.7$, $ng = 20$, (JADE: $\mu_{F,init} = 0.5$, $\mu_{CR,init} = 0.5$, $c = 0.1$, $p_{\min} = 0.05$, $p_{\max} = 0.2$), (CoDE:$\{F, CR\} \in \{\{1.0, 0.1\}, \{1.0, 0.9\}, \{0.8, 0.2\}\}$, {*rand/1/bin*, *rand/2/bin*, *current-to-rand/1*}), (EPSDE:$P_F = \{0.4, 0.5, \ldots, 0.9\}$, $P_{CR} = \{0.1, 0.2, \ldots, 0.9\}$, $P_v = \{$*rand/1*, *best/2*, *current-to-rand/1*$\}$) [59].
- GPEME, which is one of the most popular and frequently compared SAEAs, employs DE and the Kriging model. Thus, GPEME is the standard of SAEA and comparison with it is essential. We used $N = 100$, $F = 0.8$, $CR = 0.8$, $\tau = 100$, $\lambda = 50$, $l = 4$, $\omega = 2$, regression = zero-order, correlation = Gaussian, $\theta \in [10^{-5}, 10^2]$, and $\theta_{init} = 10^{-2}$ [26].
- S-JADE, which is a state-of-the-art SAEA, consists of modified JADE with multiple RBFN models. By comparing with S-JADE, we compare the performances of EBADE and state-of-the-art SAEAs with DE parameter configuration adaptation in moderately EOPs. We used $N = 100$, $F_{out} = 0.5$, $CR_{out} = 0.75$, $p_{pbest_{out}} = 0.05$, $F_{in} = 0.5$, $CR_{out} = 0.5$, $p_{pbest_{in}} = 0.1$, $std_F = 0.1$, $std_{CR} = 0.1$, $L = 10$, $\epsilon = 0.01$, $c = 0.1$, $FE_{max_{in}} = 2,000$, kernel = cubic, and $r = rand(0,\ 1.25)$ [7].
- SAHO, which is also a state-of-the-art SAEA, employs DE and TLBO as optimizers and the RBFN model as a surrogate. In addition to SAHO being one of the state-of-the-art SAEAs, comparison with GPEME and SAHO provides a relative performance comparison of EBADE for different ML models used in these SAEAs. We used $N = 100$, $F = 0.9$, $CR = 0.5$, $K = 30$, $neighbor = 5D$, and kernel = cubic [40].
- ESMDE, which adapts DE configuration during screening of the solution by the Kriging model. The candidate

configurations are two mutation strategies (*rand/1* and *current-to-rand/1*), two crossover strategies (*binomial* and *exponential*), $F \in [0.5, 1.0]$, and $CR \in [0.0, 1.0]$. By comparing with ESMDE, we verify if the adaptation mechanism of EBADE outperforms SAEAs with DE parameter adaptation in moderately EOPs. We used $c = 10$, regression = zero-order, correlation = Gaussian, $\theta \in [10^{-5}, 10^{2}]$, and $\theta_{init} = 10^{-2}$ [34].

Parameter configuration of EBADE

The size of each subpopulation was taken to be $N = 4$, the number of subpopulations was $M = 25$, the number of candidate configurations was $K = 6$, and the parameter in the *current-to-pbest/1* is $p = 0.5$. Thus, the size of the whole population $\mathcal{P}_{all}$ is $N \times M = 100$, and these configurations are identical to those in the compared adaptive DEs above. Note that δ_C in Eq. (1) was not needed, because the objective values except at the global optimum were non-negative. The objective value at the global optimum of each function was 0, but no trial reached the global optimum.

Evaluation scheme

All algorithms were forcibly terminated when the number of FEs reached its maximum budget $FE_{\max}$, including FEs used for the initialization phase. The performance of algorithms was evaluated in terms of the best objective value discovered at $FE_{\max}$, and their mean values over 21 independent trials were reported. The Wilcoxon signed-rank test was applied to identify significant differences with a significance level of $p < 0.05$. Additionally, the average runtimes of all algorithms to complete one trial were compared.

We set $FE_{\max} = 6000$ as a default value as moderately restricted budgets of FEs were assumed. However, additional results corresponding to 2000–10,000 FEs are presented in Sect. 4.3 to investigate the scalability of EBADE with respect to the number of FEs.

Results

Tables 3, 4, and 5 summarize the performances recorded corresponding to 6000 FEs for problems with $D \in \{10, 20, 30\}$, respectively. The best and worst performances are highlighted in bold and italic, respectively. In the tables, "+", "−", and "∼" indicate that the performance of a compared algorithm is statistically better than, statistically worse than, and comparable to that of EBADE, respectively. Further, the average rank and the overall statistical results, i.e., the counts of $+/-/\sim$, are summarized at the bottom of each table.

As is evident from Table 3, EBADE outperformed adaptive DEs (SHADE, jSO, CSDE, and EDEV) and one SAEA (ESMDE) significantly on multiple problem instances with $D = 10$. In particular, the performance of EBADE was statistically better than those of adaptive DEs on at least 14 problem instances. In addition, EBADE exhibited comparable performance with respect to three SAEAs (GPEME, S-JADE, and SAHO). In particular, the performance of EBADE was statistically better than SAEAs on at least nine problem instances, and there were only a maximum of six statistically worse cases. Consequently, EBADE was assigned the best rank, outperforming the considered SAEAs. This indicates a benefit of auto-tunable approaches in solving EOPs—the performance of EBADE is less problem-dependent owing to adaptive control of parameter configurations during a run. Although the performance of SAEAs could be improved via fine-tuning, it is usually hindered in EOPs.

Even when D is increased to 20 and 30, EBADE remained effective, as evidenced by Tables 4 and 5. The performance of EBADE was better than those of adaptive DEs and highly competitive with SAEAs on several problem instances. For $D = 30$, the effectiveness of EBADE was slightly poorer, as GPEME outperformed EBADE on certain problems. However, EBADE was assigned the best rank, indicating that it performed well on average. All four adaptive EAs considered for comparison were assigned worse ranks than the SAEAs, except for ESMDE, as they are not designed for restricted FEs. These observations empirically corroborate the effectiveness of our emulation-based adaptation mechanism.

Table 6 presents the average algorithmic runtimes required to complete one trial corresponding to each problem dimension. EBADE required the second-highest runtime among the five adaptive DEs, but the differences were only a few seconds at most. On the other hand, EBADE outperformed the SAEAs in terms of speed; the runtime of EBADE was smaller than that of the SAEAs by at least an order of two.

In summary, overall results suggest that, without any help of surrogate models, EBADE performs comparably with SAEAs while operating much faster than them. This observation empirically suggests that the proposed emulation-based adaptation enables the adaptive DE mechanism to accelerate its evolutionary search under a restricted number of FEs, demonstrating the possibility to realize computationally efficient and auto-tunable optimizers for solving moderately EOPs.

Additional results

To investigate the scalability of EBADE with respect to the number of FEs, we conducted additional experiments with different values of $FE_{\max}$ under the same experimental environment as in Sect. 4.1, except for $FE_{\max}$. Specifically, $FE_{\max}$ was set to {2000, 4000, 8000, 10,000} in addition to its default value of 6000. Here, we report the average ranks and the overall statistical results.

Table 3 Comparison of optimization results obtained at 6000 FEs ($D = 10$, 21 trials)

	Adaptive DEs					SAEAs			
	EBADE	SHADE	jSO	CSDE	EDEV	GPEME	S-JADE	SAHO	ESMDE
F1	4.69E–05	7.27E–01 –	1.13E–03 –	5.22E+00 –	*4.27E+02* –	**0.00E+00 +**	3.03E–13 +	1.66E–28 +	1.81E–02 –
F2	2.18E+05	3.87E+06 –	**2.24E+03 +**	*9.48E+06* –	9.20E+06 –	3.14E+06 –	1.14E+05 +	4.09E+04 +	7.40E+06 –
F3	**6.46E+03**	3.38E+06 –	7.57E+03 ∼	4.28E+07 –	8.37E+08 –	7.93E+07 –	5.07E+09 –	*2.21E+10* –	7.76E+06 –
F4	6.90E+02	1.82E+04 –	**1.09E+01 +**	*3.15E+04* –	2.52E+04 –	2.27E+04 –	9.12E+03 –	1.21E+04 –	2.48E+04 –
F5	8.49E–03	1.43E+00 –	1.15E–02 ∼	5.22E+00 –	*1.82E+02* –	**0.00E+00 +**	1.73E+00 ∼	9.17E–05 +	2.98E–01 –
F6	1.50E+01	1.02E+01 ∼	8.65E+00 ∼	1.35E+01 ∼	*5.25E+01* –	**5.81E+00** ∼	8.41E+00 ∼	7.46E+00 ∼	9.90E+00 ∼
F7	1.26E+01	2.08E+01 –	**3.83E+00 +**	4.08E+01 –	6.04E+01 –	3.54E+01 –	7.41E+01 –	*2.45E+02* –	3.67E+01 –
F8	*2.06E+01*	*2.06E+01* ∼	**2.05E+01** ∼	*2.06E+01* ∼	*2.06E+01* ∼	*2.06E+01* ∼	*2.06E+01* ∼	*2.06E+01* ∼	*2.06E+01* ∼
F9	**3.93E+00**	8.62E+00 –	7.74E+00 –	*9.66E+00* –	9.53E+00 –	4.67E+00 –	4.50E+00 ∼	5.40E+00 –	8.99E+00 –
F10	6.43E–01	5.73E+00 –	5.24E–01 ∼	2.70E+01 –	*1.32E+02* –	1.67E–01 +	**8.62E–02 +**	4.59E–01 +	3.46E+00 –
F11	**6.16E+00**	2.61E+01 –	2.25E+01 –	3.09E+01 –	*4.92E+01* –	1.49E+01 –	1.05E+01 –	2.99E+01 –	2.07E+01 –
F12	**1.83E+01**	4.30E+01 –	3.01E+01 –	5.10E+01 –	*6.74E+01* –	2.34E+01 ∼	1.88E+01 ∼	2.71E+01 –	4.73E+01 –
F13	2.38E+01	4.47E+01 –	3.20E+01 –	4.86E+01 –	*6.61E+01* –	3.42E+01 –	2.38E+01 ∼	4.80E+01 –	4.63E+01 –
F14	**1.83E+02**	1.28E+03 –	1.18E+03 –	*1.39E+03* –	1.32E+03 –	7.76E+02 –	3.47E+02 –	1.14E+03 –	1.18E+03 –
F15	**1.08E+03**	1.77E+03 –	1.64E+03 –	*1.91E+03* –	1.86E+03 –	1.21E+03 ∼	1.15E+03 ∼	1.31E+03 ∼	1.85E+03 –
F16	1.53E+00	*1.95E+00* –	1.78E+00 –	1.78E+00 ∼	1.83E+00 –	1.80E+00 –	1.58E+00 ∼	**1.20E+00 +**	1.86E+00 –
F17	**2.00E+01**	3.93E+01 –	3.97E+01 –	4.05E+01 –	*7.69E+01* –	2.31E+01 ∼	2.27E+01 ∼	2.46E+01 –	3.33E+01 –
F18	3.80E+01	5.57E+01 –	4.96E+01 –	5.87E+01 –	*8.92E+01* –	3.57E+01 ∼	3.74E+01 ∼	**2.66E+01 +**	5.69E+01 –
F19	**1.74E+00**	3.85E+00 –	2.52E+00 –	3.94E+00 –	*1.01E+01* –	4.54E+00 –	8.43E+00 –	1.84E+00 ∼	3.75E+00 –
F20	3.61E+00	3.84E+00 –	**3.45E+00** ∼	4.05E+00 –	*4.10E+00* –	3.63E+00 ∼	3.75E+00 ∼	4.04E+00 –	4.08E+00 –
F21	3.81E+02	4.00E+02 –	4.00E+02 –	4.00E+02 –	*4.43E+02* –	3.91E+02 –	4.17E+02 –	4.31E+02 ∼	3.81E+02 ∼
F22	**3.06E+02**	1.43E+03 –	1.45E+03 –	1.30E+03 –	1.45E+03 –	8.79E+02 –	4.78E+02 –	*1.50E+03* –	1.34E+03 –
F23	1.44E+03	1.97E+03 –	1.77E+03 –	*2.09E+03* –	2.06E+03 –	1.30E+03 ∼	**1.26E+03 +**	1.79E+03 –	1.93E+03 –
F24	2.13E+02	2.18E+02 –	**2.10E+02 +**	2.22E+02 –	*2.24E+02* –	2.14E+02 ∼	2.14E+02 ∼	2.17E+02 –	2.22E+02 –
F25	2.13E+02	2.20E+02 –	**2.11E+02** ∼	*2.24E+02* –	2.23E+02 –	2.15E+02 ∼	2.14E+02 ∼	2.19E+02 –	2.23E+02 –
F26	1.65E+02	1.78E+02 ∼	1.94E+02 ∼	*2.01E+02* –	1.98E+02 –	1.85E+02 ∼	**1.53E+02** ∼	1.97E+02 –	1.93E+02 –
F27	**4.23E+02**	4.87E+02 –	4.44E+02 ∼	5.66E+02 –	*5.81E+02* –	5.03E+02 –	4.35E+02 ∼	5.16E+02 –	5.45E+02 –
F28	4.12E+02	3.55E+02 ∼	3.01E+02 +	4.13E+02 ∼	6.40E+02 –	**2.90E+02 +**	7.21E+02 –	*1.08E+03* —	3.27E+02 ∼
+/ − / ∼		0/24/4	5/14/9	0/24/4	0/27/1	4/13/11	4/9/15	6/17/5	0/24/4
Ave. rank	**2.45**	5.59	3.62	7.41	8.23	3.46	3.59	4.88	5.77

Table 4 Comparison of optimization results obtained at 6000 FEs ($D = 20$, 21 trials)

	Adaptive DEs					SAEAs			
	EBADE	SHADE	jSO	CSDE	EDEV	GPEME	S-JADE	SAHO	ESMDE
F1	1.49E+00	4.30E+01 -	1.62E+01 -	2.05E+02 -	*3.60E+03* -	3.80E+01 -	2.50E-05 +	**3.30E-27 +**	3.21E+01 -
F2	6.04E+06	3.83E+07 -	1.89E+06 +	8.38E+07 -	*1.03E+08* -	3.18E+07 -	2.62E+06 +	**1.77E+05 +**	6.74E+07 -
F3	**1.31E+09**	1.82E+09 ∼	1.89E+09 ∼	1.60E+10 -	5.69E+10 -	2.34E+10 -	6.20E+12 -	*8.99E+14* -	2.87E+10 -
F4	1.42E+04	4.99E+04 -	**6.90E+03 +**	*8.91E+04* -	6.88E+04 -	7.60E+04 -	3.68E+04 -	3.98E+04 -	5.91E+04 -
F5	2.43E+01	4.52E+01 -	4.09E+01 -	1.50E+02 -	*2.57E+03* -	3.08E+02 -	6.69E+01 -	**1.16E-04 +**	7.17E+01 -
F6	4.63E+01	6.40E+01 ∼	4.40E+01 ∼	1.30E+02 -	*5.88E+02* -	1.66E+01 +	1.05E+00 +	**2.78E-02 +**	1.14E+02 -
F7	8.57E+01	7.10E+01 ∼	**5.61E+01 +**	1.28E+02 -	1.99E+02 -	1.42E+02 -	1.54E+03 -	*1.52E+04* -	1.49E+02 -
F8	2.09E+01	2.09E+01 ∼	*2.10E+01* -	*2.10E+01* ∼	*2.10E+01* ∼	*2.10E+01* -	2.09E+01 ∼	*2.10E+01* -	2.09E+01 ∼
F9	1.63E+01	2.41E+01 -	2.26E+01 -	*2.57E+01* -	2.56E+01 -	**1.31E+01 +**	1.66E+01 ∼	1.85E+01 -	2.40E+01 -
F10	3.66E+01	6.99E+01 -	7.81E+00 +	2.99E+02 -	*9.96E+02* -	2.04E+01 ∼	7.43E-01 +	**1.07E-01 +**	2.72E+02 -
F11	4.94E+01	1.06E+02 -	1.14E+02 -	1.12E+02 -	*1.90E+02* -	**4.67E+01** ∼	5.11E+01 ∼	9.82E+01 -	9.54E+01 -
F12	8.04E+01	1.37E+02 -	1.18E+02 -	1.58E+02 -	*2.34E+02* -	**5.76E+01 +**	7.16E+01 ∼	8.87E+01 ∼	1.52E+02 -
F13	1.16E+02	1.34E+02 -	1.25E+02 ∼	1.55E+02 -	*2.34E+02* -	1.08E+02 ∼	**1.02E+02 +**	1.33E+02 ∼	1.58E+02 -
F14	**1.55E+03**	3.63E+03 -	*3.69E+03* -	3.42E+03 -	3.52E+03 -	2.18E+03 -	2.07E+03 -	2.85E+03 -	3.05E+03 -
F15	3.73E+03	4.68E+03 -	4.74E+03 -	*5.00E+03* -	4.78E+03 -	3.70E+03 ∼	3.66E+03 ∼	**3.04E+03 +**	4.73E+03 -
F16	2.59E+00	2.74E+00 ∼	*2.94E+00* -	2.71E+00 ∼	2.80E+00 ∼	2.88E+00 ∼	2.83E+00 ∼	**2.19E+00 +**	2.92E+00 -
F17	9.93E+01	1.33E+02 -	1.40E+02 -	1.47E+02 -	*3.03E+02* -	**6.82E+01 +**	8.64E+01 +	6.88E+01 +	1.28E+02 -
F18	1.48E+02	1.65E+02 -	1.53E+02 ∼	1.80E+02 -	*2.96E+02* -	1.40E+02 ∼	1.16E+02 +	**8.10E+01 +**	1.70E+02 -
F19	**1.09E+01**	1.24E+01 -	1.15E+01 ∼	2.08E+01 -	*1.26E+03* -	1.26E+01 -	5.15E+01 -	1.78E+01 -	2.45E+01 -
F20	9.49E+00	9.94E+00 -	9.26E+00 ∼	9.87E+00 -	9.92E+00 -	**9.22E+00** ∼	9.96E+00 -	9.93E+00 -	*9.99E+00* -
F21	**3.84E+02**	4.52E+02 -	4.35E+02 -	7.57E+02 -	1.06E+03 -	4.11E+02 ∼	1.16E+03 -	*2.16E+03* -	5.56E+02 -
F22	**1.98E+03**	3.93E+03 -	*4.39E+03* -	3.87E+03 -	4.00E+03 -	2.87E+03 -	2.07E+03 ∼	3.47E+03 -	3.67E+03 -
F23	4.20E+03	5.28E+03 -	5.21E+03 -	*5.36E+03* -	5.24E+03 -	3.62E+03 ∼	4.26E+03 ∼	**3.59E+03 +**	5.27E+03 -
F24	2.43E+02	2.58E+02 -	2.46E+02 ∼	2.65E+02 -	*2.68E+02* -	**2.38E+02** ∼	2.55E+02 -	2.54E+02 -	2.63E+02 -
F25	2.53E+02	2.62E+02 -	2.53E+02 ∼	2.71E+02 -	*2.72E+02* -	**2.48E+02 +**	2.66E+02 -	2.64E+02 -	2.67E+02 -
F26	2.26E+02	2.05E+02 ∼	**2.00E+02 +**	2.25E+02 ∼	2.16E+02 ∼	*2.31E+02* -	2.03E+02 ∼	2.20E+02 ∼	2.10E+02 ∼
F27	7.23E+02	8.70E+02 -	7.83E+02 -	9.54E+02 -	*9.55E+02* -	**6.34E+02 +**	7.09E+02 ∼	7.74E+02 ∼	9.13E+02 -
F28	1.48E+03	1.56E+03 ∼	**1.40E+03** ∼	1.96E+03 -	2.67E+03 -	2.56E+03 -	3.06E+03 -	*4.80E+03* -	2.06E+03 -
+/ − / ∼		0/21/7	5/14/9	0/25/3	0/25/3	6/12/10	7/11/10	10/14/4	0/26/2
Ave. rank	**2.82**	5.20	4.30	6.93	7.86	3.68	4.02	4.07	6.12

Table 5 Comparison of optimization results obtained at 6000 FEs ($D = 30$, 21 trials)

	Adaptive DEs					SAEAs			
	EBADE	SHADE	jSO	CSDE	EDEV	GPEME	S-JADE	SAHO	ESMDE
F1	1.59E+02	2.39E+02 -	3.27E+02 -	1.06E+03 -	*9.01E+03* -	5.44E+02 ∼	3.08E-07 +	**7.24E-27 +**	6.34E+02 -
F2	2.96E+07	9.54E+07 -	1.63E+07 +	2.31E+08 -	*3.00E+08* -	2.36E+07 ∼	5.74E+06 +	**4.77E+05 +**	1.87E+08 -
F3	1.07E+10	**1.00E+10** ∼	1.88E+10 -	5.76E+10 -	1.13E+11 -	6.43E+10 -	1.95E+11 -	*1.03E+15* -	1.01E+11 -
F4	**3.71E+04**	9.63E+04 -	3.85E+04 ∼	*1.32E+05* -	1.26E+05 -	1.29E+05 -	6.56E+04 -	6.95E+04 -	1.17E+05 -
F5	1.95E+02	2.10E+02 ∼	5.69E+02 -	7.27E+02 -	*8.92E+03* -	9.42E+02 ∼	2.07E+02 ∼	**1.87E-04 +**	6.50E+02 -
F6	1.37E+02	1.28E+02 ∼	1.34E+02 ∼	2.23E+02 -	*1.11E+03* -	3.88E+01 +	4.37E+01 +	**1.43E+01 +**	2.23E+02 -
F7	1.56E+02	1.30E+02 +	**1.29E+02 +**	1.94E+02 -	2.78E+02 -	2.29E+02 -	2.35E+02 -	*7.76E+03* -	2.37E+02 -
F8	**2.11E+01**	**2.11E+01** ∼	**2.11E+01** ∼	**2.11E+01** ∼	**2.11E+01** ∼	**2.11E+01** ∼	**2.11E+01** ∼	**2.11E+01** ∼	**2.11E+01** ∼
F9	3.11E+01	4.00E+01 -	4.09E+01 -	*4.27E+01* -	4.19E+01 -	**2.28E+01 +**	2.56E+01 +	3.24E+01 ∼	3.98E+01 -
F10	1.52E+02	1.94E+02 -	1.04E+02 +	8.26E+02 -	*2.57E+03* -	1.77E+02 ∼	1.31E+00 +	**5.18E-02 +**	8.63E+02 -
F11	1.18E+02	2.04E+02 -	2.25E+02 -	2.18E+02 -	*3.99E+02* -	**9.87E+01** ∼	1.50E+02 -	2.68E+02 -	2.17E+02 -
F12	1.97E+02	2.46E+02 -	2.27E+02 -	2.73E+02 -	*4.50E+02* -	**1.23E+02 +**	1.85E+02 ∼	2.41E+02 ∼	2.93E+02 -
F13	2.65E+02	2.47E+02 ∼	2.39E+02 +	2.77E+02 ∼	*4.27E+02* -	**2.18E+02 +**	2.34E+02 +	2.54E+02 ∼	2.93E+02 -
F14	**3.75E+03**	6.81E+03 -	*7.73E+03* -	6.60E+03 -	6.56E+03 -	4.04E+03 ∼	4.34E+03 ∼	4.48E+03 -	6.13E+03 -
F15	7.04E+03	8.14E+03 -	8.27E+03 -	8.36E+03 -	*8.43E+03* -	7.79E+03 -	6.83E+03 ∼	**5.12E+03 +**	8.14E+03 -
F16	3.24E+00	3.52E+00 ∼	3.65E+00 -	3.63E+00 -	*3.74E+00* -	3.48E+00 ∼	3.71E+00 -	**3.13E+00** ∼	3.68E+00 -
F17	2.00E+02	2.53E+02 -	2.70E+02 -	2.98E+02 -	*5.84E+02* -	**1.31E+02 +**	1.76E+02 ∼	1.40E+02 +	2.89E+02 -
F18	2.97E+02	2.95E+02 ∼	2.77E+02 +	3.32E+02 -	*6.01E+02* -	2.76E+02 +	2.18E+02 +	**1.40E+02 +**	3.37E+02 -
F19	3.24E+01	2.63E+01 ∼	2.37E+01 +	1.06E+02 -	*4.05E+04* -	1.99E+01 +	1.27E+02 -	**1.44E+01 +**	4.67E+02 -
F20	1.44E+01	1.49E+01 -	1.43E+01 ∼	1.48E+01 -	1.48E+01 -	**1.38E+01 +**	1.46E+01 ∼	1.48E+01 -	*1.50E+01* -
F21	**6.47E+02**	7.05E+02 ∼	8.23E+02 -	1.34E+03 -	2.22E+03 -	2.20E+03 -	2.38E+03 -	*3.59E+03* -	1.38E+03 -
F22	**3.92E+03**	7.37E+03 -	*8.22E+03* -	7.52E+03 -	7.70E+03 -	4.55E+03 ∼	5.46E+03 -	5.27E+03 -	6.87E+03 -
F23	7.58E+03	8.82E+03 -	8.76E+03 -	*8.87E+03* -	8.71E+03 -	7.51E+03 ∼	7.33E+03 ∼	**5.68E+03 +**	8.72E+03 -
F24	2.79E+02	2.92E+02 -	2.86E+02 -	3.04E+02 -	*3.12E+02* -	**2.62E+02 +**	2.85E+02 -	2.82E+02 ∼	3.03E+02 -
F25	2.97E+02	3.13E+02 -	3.11E+02 -	3.19E+02 -	*3.23E+02* -	**2.79E+02 +**	3.04E+02 -	3.00E+02 ∼	3.16E+02 -
F26	2.74E+02	2.72E+02 ∼	**2.17E+02** ∼	*3.50E+02* -	3.07E+02 ∼	3.22E+02 ∼	2.35E+02 ∼	3.05E+02 ∼	2.52E+02 ∼
F27	1.04E+03	1.26E+03 -	1.23E+03 -	1.36E+03 -	*1.37E+03* -	**8.88E+02 +**	9.96E+02 ∼	1.07E+03 ∼	1.32E+03 -
F28	1.34E+03	**1.14E+03** ∼	1.46E+03 ∼	1.87E+03 -	2.93E+03 -	2.13E+03 -	3.07E+03 -	*6.38E+03* -	2.08E+03 -
+/ − / ∼		1/16/11	6/16/6	0/26/2	0/26/2	11/6/11	7/11/10	10/9/9	0/26/2
Ave. rank	**3.18**	4.84	4.57	6.88	8.04	3.43	3.79	3.86	6.43

Table 6 Average runtime [sec] required to complete one trial ($FE_{\max} = 6000$)

D	EBADE	SHADE	jSO	CSDE	EDEV	GPEME	S-JADE	SAHO	ESMDE
10	1.42E+01	1.18E+01	6.95E+00	1.35E+01	1.58E+01	3.30E+03	1.04E+05	4.13E+04	2.98E+03
20	1.45E+01	1.18E+01	6.98E+00	1.37E+01	1.80E+01	1.54E+04	1.11E+05	7.71E+04	1.53E+04
30	1.55E+01	1.21E+01	7.23E+00	1.44E+01	1.68E+01	3.34E+04	7.72E+04	8.65E+04	3.67E+04

Table 7 reports the statistical results summarized as the counts of $+/-/\sim$. Figure 1 depicts the average ranks of all nine algorithms. Overall, the performance of EBADE was statistically better than those of all the four adaptive DEs even when $FE_{\max}$ was decreased/increased from its default value 6000. The superiority of EBADE over the four adaptive DEs is also corroborated by Fig. 1, since EBADE was assigned the best rank among five adaptive DEs corresponding to all problem dimensions and all $FE_{\max}$ cases. These results prove that EBADE was successfully adapted to EOPs.

When $FE_{\max}$ was 2000, three SAEAs—GPEME, S-JADE, and SAHO—outperformed EBADE corresponding to all problem dimensions, as the number of "+" was sufficiently larger than that of "−". This result clearly demonstrated the effectiveness of surrogate-assisted search on highly restricted FE budgets. However, when $FE_{\max}$ was increased to 4000, the performance of EBADE became competitive to those of the SAEAs, except in the 30-dimensional cases. When $FE_{\max}$ was further increased to 8000 and 10,000, the effectiveness of EBADE was highlighted even more. These tendencies can be also observed in Fig. 1—EBADE was always assigned the second or third rank when $FE_{\max} \leq 4000$, and its rank improved to first for $FE_{\max} \geq 6000$. On the other hand, the ranks of SAEAs gradually decreased with the increase of $FE_{\max}$; this indicates stagnation in SAEA performances. This is because SAEAs tend to struggle with premature convergence with a higher number of FEs. As observed in [47, 60], generally, surrogate-assisted searches provide a strong exploitation bias to promote evolutionary search under very restricted budgets of FEs. However, this degrades the diversity of solutions with a higher number of FEs, degrading the search performance and the quality of (global) surrogates. Consequently, as shown in Fig. 1, SAEAs performed very well for very restricted budgets of FEs; however, their performances gradually degraded with the increase of FEs. EBADE was assigned to the best rank for all dimensions when $FE_{\max} \geq 6000$, as shown in Fig. 1. Thus, this observation indicates that EBADE is well scaled to the increase of the problem dimensions D on moderately EOPs. A possible reason for this is that EBADE can possess a better diversity of solutions than that of SAEAs; after the adaptation of parameter configurations, EBADE generates solutions similar to the standard DEs without any pre-screening process using surrogates.

The following differences were identified between the different algorithmic mechanisms. The relatively simple algorithm of SHADE ensured its scalability with respect to D, since the average rank increased slightly as D increased, as in Fig. 1. jSO also exhibited improved performance—it ranked second among adaptive DEs and its average rank was slightly higher than those of all SAEAs when $FE_{\max} \geq 8000$ ($D = 10$) and $FE_{\max} = 10{,}000$ ($D = 20$), as in Fig. 1. This was attributed to the mechanism by which jSO reduces the population size towards each set $FE_{\max}$ to gain the number of solution evolution. If this mechanism were incorporated into EBADE, it may further improve its performance. However, EBADE already outperformed SHADE and jSO based solely on the prior validation mechanism and subpopulation-based adaptation. As depicted in Fig. 1, the average rank of CSDE increased with an increase in $FE_{\max}$ for all dimensions, but the statistical test results obtained in comparison with EBADE did not improve, as presented in Table 7. Thus, the state-of-the-art CSDE is specialized for cases with plentiful FE budgets. EDEV seemed to consume a large number of FEs to improve performance, because it has a huge search space of parameter configurations. In comparison with SAEAs, EBADE was highly competitive beyond the type of ML used in compared SAEAs, since its rank was higher than those of GPEME and SAHO, at least for $FE_{\max} \geq 6000$, as in Fig. 1. Although S-JADE and ESMDE adaptively control DE parameter configurations, further consideration may be required in the combined implementation of SAEAs and adaptation methods to solve moderately EOPs effectively, especially owing to the low rank of ESMDE in Fig. 1.

In summary, the compared SAEAs exhibited excellent performances when $FE_{\max} = 2000$, but EBADE derived good performances when $FE_{\max} \geq 6000$. This observation shows the effectiveness of EBADE on moderately EOPs.

Discussion

In this section, the results of ablation studies are reported to confirm the effectiveness of the main components of EBADE. First, we discuss the effect of controlling the DE parameter configurations in moderately EOPs. Subsequently, the effectiveness of the prior validation and subpopulation-based adaptation is discussed. Finally, the adaptation results of

Table 7 Statistical results (the count of $+/-/\sim$) for 2000, 4000, 6000, 8000, and 10,000 FEs

a) $D = 10$

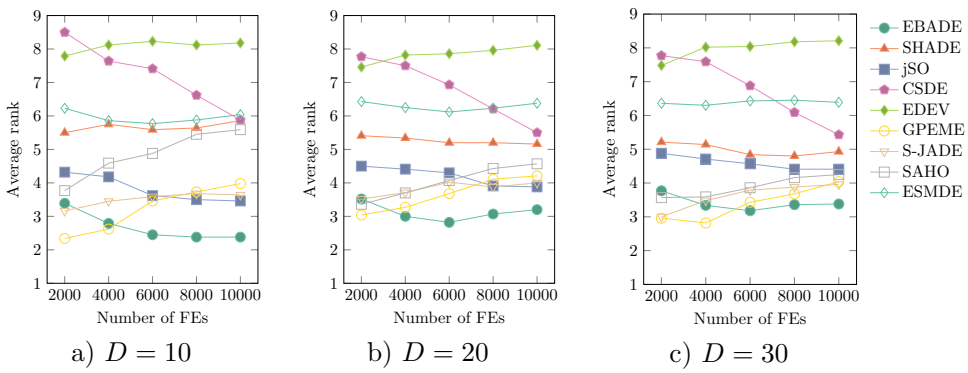
FEs	vs SHADE	vs jSO	vs CSDE	vs EDEV	vs GPEME	vs S-JADE	vs SAHO	vs ESMDE
2000	0/24/ 4	7/ 9/12	0/26/ 2	0/26/ 2	17/ 6/ 5	14/ 4/10	10/ 5/13	0/25/ 3
4000	0/23/ 5	5/15/ 8	0/26/ 2	0/26/ 2	7/ 9/12	5/ 8/15	7/13/ 8	0/24/ 4
6000	0/24/ 4	5/14/ 9	0/24/ 4	0/27/ 1	4/13/11	4/ 9/15	6/17/ 5	0/24/ 4
8000	1/24/ 3	6/13/ 9	1/24/ 3	0/26/ 2	4/16/ 8	3/ 9/16	2/18/ 8	0/24/ 4
10,000	0/24/ 4	6/13/ 9	0/25/ 3	0/26/ 2	4/18/ 6	3/ 9/16	3/19/ 6	0/24/ 4

b) $D = 20$

FEs	vs SHADE	vs jSO	vs CSDE	vs EDEV	vs GPEME	vs S-JADE	vs SAHO	vs ESMDE
2000	0/22/ 6	0/ 8/20	0/25/ 3	0/24/ 4	18/ 6/ 4	16/ 6/ 6	18/ 6/ 4	0/24/ 4
4000	0/24/ 4	4/13/11	0/25/ 3	0/25/ 3	12/ 8/ 8	8/10/10	11/12/ 5	0/25/ 3
6000	0/21/ 7	5/14/ 9	0/25/ 3	0/25/ 3	6/12/10	7/11/10	10/14/ 4	0/26/ 2
8000	1/21/ 6	7/14/ 7	0/23/ 5	0/25/ 3	4/11/13	6/11/11	8/16/ 4	0/26/ 2
10,000	2/19/ 7	8/14/ 6	1/21/ 6	0/25/ 3	3/10/15	6/10/12	7/16/ 5	0/25/ 3

c) $D = 30$

FEs	vs SHADE	vs jSO	vs CSDE	vs EDEV	vs GPEME	vs S-JADE	vs SAHO	vs ESMDE
2000	0/22/ 6	0/13/15	0/26/ 2	1/25/ 2	18/ 6/ 4	17/ 5/ 6	18/ 6/ 4	0/25/ 3
4000	1/18/ 9	3/14/11	0/26/ 2	0/26/ 2	16/ 6/ 6	13/ 9/ 6	12/ 8/ 8	0/25/ 3
6000	1/16/11	6/16/ 6	0/26/ 2	0/26/ 2	11/ 6/11	7/11/10	10/ 9/ 9	0/26/ 2
8000	5/13/10	5/14/ 9	1/23/ 4	0/25/ 3	10/ 9/ 9	8/12/ 8	10/13/ 5	0/26/ 2
10,000	7/14/ 7	7/14/ 7	1/22/ 5	0/25/ 3	7/10/11	6/13/ 9	9/13/ 6	0/24/ 4



a) $D = 10$ b) $D = 20$ c) $D = 30$

Fig. 1 Average rank of all nine algorithms over the number of FEs

the DE parameter configurations are investigated. Experimental environments identical to those reported in Sect. 4.1 are used with $FE_{\max} \in \{2000, 4000, 6000, 8000, 10{,}000\}$ unless stated otherwise.

Impact of parameter adaptation in moderately EOPs

We empirically demonstrate that controlling the DE parameter configuration is crucial to improve the performance of adaptive DEs even for moderately EOPs. To this end, we compared the performances of EBADE and various DEs with fixed parameter configurations. In particular, we introduced eight DE variants with different combinations of mutation and crossover strategies; *best/1/bin* (b/1/b), *current-to-best/1/bin* (cb/1/b), *current-to-pbest/1/bin* (cpb/1/b), *rand-to-best/1/bin* (rb/1/b), *best/1/exp* (b/1/e), *current-to-best/1/exp* (cb/1/e), *current-to-pbest/1/exp* (cpb/1/e), and *rand-to-best/1/exp* (rb/1/e), where *bin* and *exp* denote *binomial* and *exponential* crossover, respectively; and $F = 0.5$ and $CR = 0.9$ were used for all variants. For cpb/1/b and cpb/1/e, we used $p = 0.5$.

Table 8 summarizes the statistical results in terms of the counts of $+/-/\sim$, where "+", "−", and "∼" indicate that the performance of a DE variant is statistically better than, statistically worse than, and comparable with that of EBADE, respectively. Further, Fig. 2 depicts the average ranks. In Table 8, two DE variants using cb/1/b and rb/1/b were observed to outperform EBADE on several prob-

Table 8 Statistical results (the count of $+/-/\sim$) for 2000, 4000, 6000, 8000, and 10,000 FEs in comparison with eight DE variants with fixed parameter configurations

a) $D = 10$								
FEs	vs b/1/b	vs cb/1/b	vs cpb/1/b	vs rb/1/b	vs b/1/e	vs cb/1/e	vs cpb/1/e	vs rb/1/e
2000	8/ 6/14	8/ 8/12	0/17/11	12/ 8/ 8	4/ 7/17	0/13/15	0/25/ 3	2/13/13
4000	3/12/13	7/11/10	1/15/12	11/10/ 7	5/13/10	0/17/11	0/22/ 6	0/16/12
6000	1/14/13	7/11/10	2/16/10	11/10/ 7	4/16/ 8	0/18/10	0/22/ 6	3/18/ 7
8000	0/13/15	8/14/ 6	4/16/ 8	9/14/ 5	5/16/ 7	2/18/ 8	0/21/ 7	4/19/ 5
10,000	0/13/15	8/14/ 6	5/16/ 7	9/13/ 6	5/16/ 7	2/18/ 8	0/21/ 7	4/17/ 7
b) $D = 20$								
FEs	vs b/1/b	vs cb/1/b	vs cpb/1/b	vs rb/1/b	vs b/1/e	vs cb/1/e	vs cpb/1/e	vs rb/1/e
2000	7/ 8/13	11/ 3/14	0/17/11	15/ 3/10	2/13/13	0/22/ 6	0/24/ 4	0/16/12
4000	2/18/ 8	5/10/13	1/15/12	11/ 7/10	1/14/13	0/19/ 9	0/25/ 3	1/17/10
6000	1/18/ 9	7/14/ 7	2/15/11	7/11/10	3/14/11	1/16/11	0/25/ 3	2/14/12
8000	1/22/ 5	6/17/ 5	3/16/ 9	7/16/ 5	6/14/ 8	2/14/12	0/24/ 4	7/15/ 6
10,000	1/21/ 6	6/18/ 4	5/19/ 4	6/17/ 5	6/14/ 8	3/15/10	1/23/ 4	7/16/ 5
c) $D = 30$								
FEs	vs b/1/b	vs cb/1/b	vs cpb/1/b	vs rb/1/b	vs b/1/e	vs cb/1/e	vs cpb/1/e	vs rb/1/e
2000	6/11/11	12/ 4/12	0/16/12	15/ 3/10	0/19/ 9	0/24/ 4	0/25/ 3	0/24/ 4
4000	2/19/ 7	3/10/15	2/14/12	8/ 6/14	0/21/ 7	0/24/ 4	0/25/ 3	0/22/ 6
6000	2/22/ 4	6/19/ 3	3/17/ 8	7/15/ 6	0/16/12	0/19/ 9	0/26/ 2	2/16/10
8000	1/22/ 5	5/19/ 4	4/17/ 7	5/17/ 6	1/15/12	1/18/ 9	0/25/ 3	3/16/ 9
10,000	1/22/ 5	5/19/ 4	3/16/ 9	5/17/ 6	1/14/13	2/16/10	0/21/ 7	3/12/13

lem instances, with $FE_{\max} \leq 4000$ as a sufficiently large number of "+". However, when $FE_{\max}$ was increased to 6000, the performance of EBADE was gradually improved. When $FE_{\max}$ was further increased to 8000 and 10,000, EBADE statistically outperformed all DE variants on at least 13 problem instances. Almost an identical trend was observed in Fig. 2. Specifically, although the average rank of EBADE was sometimes below those of cb/1/b and rb/1/b for $FE_{\max} \leq 4000$, EBADE always ranked first for $FE_{\max} \geq 6000$.

These results indicate that the configurations of cb/1/b and rb/1/b, i.e., providing a strong bias for exploitation, enhanced DE performance when $FE_{\max}$ was highly restricted; however, this benefit may be less important as SAEAs performed well under such restricted budgets. An important drawback of using these settings was observed to be premature convergence when $FE_{\max}$ was increased to, for example, 6000 or even 10,000. Actually, the average ranks of cb/1/b and rb/1/b were rapidly degraded as $FE_{\max}$ was increased, as depicted in Fig. 2. In contrast, EBADE performed well by controlling the DE parameter configurations, especially for $FE_{\max} \geq 6000$, i.e., under moderately restricted FE budgets.

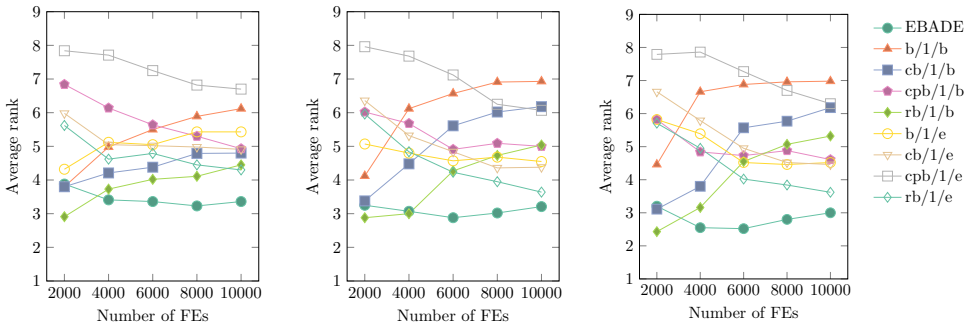
General trends of fixed DE variants are as follows. When $FE_{\max}$ was increased from 2000 to 10,000, the average ranks of DE variants using the *binomial* crossover strategy, except for cpb/1/b, i.e., b/1/b, cb/1/b, and rb/1/b, continued to decrease, while those of cpb/1/b or DE variants using the *exponential* crossover strategy, i.e., b/1/e, cb/1/e, cpb/1/e, and rb/1/e, continued to improve. EBADE was assigned consistently to good ranks. This result shows that the adaptation of the DE parameter configurations was effective in solving moderately EOPs.

Impact of adaptation of DE parameters

EBADE is designed to simultaneously adapt the DE parameter configurations (F, CR, and the mutation and crossover strategies) to determine good parameter configurations from a variety of candidates. We further validate the effectiveness of this strategy by comparing EBADE with its variant using fixed values of θ_F and θ_{CR}, denoted as EBADE-fixFCR.

Specifically, EBADE-fixFCR is designed to always use $\theta_F = 0.5$ and $\theta_{CR} = 0.9$ [46]; it tunes only the mutation and crossover strategies. Table 9 presents the statistical results summarized as the counts of $+/-/\sim$ with $FE_{\max} \in \{2000, 4000, 6000, 8000, 10{,}000\}$. In the table, "+", "−", and "∼" indicate that the performance of EBADE-fixFCR is statistically better than, statistically worse than, and comparable to that of EBADE, respectively.

When $FE_{\max}$ was 2000, EBADE-fixFCR statistically outperformed EBADE on more than seven problems for all problem dimensions. This indicates that the recommended values of θ_F and θ_{CR} boost the convergence speed of EBADE-fixFCR under the restricted budgets of FEs, by reducing



a) $D = 10$ b) $D = 20$ c) $D = 30$

Fig. 2 Average ranks of fixed DE variants and EBADE over the number of FEs

Table 9 Statistical results (the count of $+/-/\sim$) for 2000, 4000, 6000, 8000, and 10,000 FEs in comparison with EBADE and a variant where θ_F and θ_{CR} are fixed to 0.5 and 0.9, respectively

FEs	$D = 10$	$D = 20$	$D = 30$
2000	8/ 4/16	11/ 3/14	11/ 4/13
4000	9/11/ 8	10/ 6/12	10/ 6/12
6000	8/12/ 8	7/ 8/13	6/10/12
8000	10/14/ 4	6/12/10	5/10/13
10,000	10/13/ 5	6/14/ 8	5/12/11

the search space of parameter configurations. However, with $FE_{\max} \geq 6000$, the number of "−" was larger than that of "+" for all D. This is highlighted particularly for $D = 20$ and 30. For example, EBADE outperformed EBADE-fixFCR on 12 problems with $D = 20$, $FE_{\max} = 8000$ while reducing the number of "+". Consequently, the adaptation of DE parameter configurations including θ_F and θ_{CR} is important to enhance the performance of EBADE in moderately EOPs.

Parameter analysis for the number of candidate configurations

Now, we evaluate the effectiveness of the prior validation process. To this end, we introduce an EBADE variant without the prior validation process, which can be implemented by setting K to 1. EBADE with $K = 1$ always generates only one candidate parameter configuration, and thus randomly selects a parameter configuration. Further, a sensitivity analysis is performed for K. In particular, we evaluate the performance of EBADE with $K \in \{1, 2, 4, 8, 10\}$ in addition to its default value, six.

Table 10 summarizes the statistical results of the counts of $+/-/\sim$, where "+", "−", and "∼" indicate that the performance of an EBADE variant is statistically better than, statistically worse than, and comparable with that of EBADE with the default value $K = 6$, respectively. To denote the variant without the prior validation process, "(w/o PV)" is appended to $K = 1$ in the table. As shown in the table, the performance of EBADE was sensitive to the parameter K, because the statistical results changed dramatically as K shifted. In particular, when prior validation was excluded ($K = 1$), the performance of EBADE degraded significantly on multiple problem instances compared to EBADE using the default value $K = 6$, confirming the effectiveness of the prior validation process. Moreover, $K = 2$ was inappropriate, since EBADE with the default setting ($K = 6$) statistically outperformed that with $K = 2$. Thus, two candidates of parameter configurations are insufficient to conduct prior validation efficiently. When the value of K increased to 10, the performance of EBADE slightly improved for $FE_{\max} \leq 4000$; however, it degraded for $FE_{\max} \geq 8000$. This is because a large value of K tends to increase the exploitation bias in the search, and thereby EBADE may suffer from premature convergence. Specifically, when the value of K increases, a greater variety of parameter configurations is sampled and then EBADE tends to select DE parameter configurations that generate overly similar solutions to the target solution.

Parameter analysis for the number of subpopulations

We also validate the effectiveness of subpopulation-based adaptation. For this purpose, an EBADE variant that adapts DEs for individual-based adaptation is prepared. This variant can be implemented by setting $M = 100$ and $N = 1$. The size of the whole population of this variant is the same as the original EBADE; $N \times M = 4 \times 25 = 100$ and $N \times M = 1 \times 100 = 100$ for EBADE and this variant, respectively. Thus, they can be fairly compared. The number of top solutions in the post hoc validation phase, described in line 20 in Algorithm 4, is set to 25 for this variant, as all parameter configurations remain constant throughout the search stage otherwise. Additionally, we conduct a sensitivity analysis for M. Specifically, we evaluate the performance of EBADE with $M \in \{2, 4, 5, 10, 20, 50, 100\}$ in addition to its default value 25. For these M, N was set to $N \in \{50, 25, 20, 10, 5, 2, 1\}$, respectively, to ensure that the population size remains 100.

Table 10 Statistical results (the count of $+/-/\sim$) for 2000, 4000, 6000, 8000, and 10,000 FEs in comparison with EBADE with $K \in \{1, 2, 4, 8, 10\}$

a) $D = 10$					
FEs	vs $K = 1$ (w/o PV)	vs $K = 2$	vs $K = 4$	vs $K = 8$	vs $K = 10$
2000	0/16/12	0/13/15	0/ 2/26	1/ 0/27	3/ 0/25
4000	0/17/11	0/12/16	0/ 7/21	2/ 1/25	5/ 1/22
6000	0/16/12	0/13/15	1/ 5/22	2/ 2/24	3/ 3/22
8000	1/15/12	1/11/16	2/ 6/20	4/ 2/22	3/ 3/22
10,000	2/10/16	2/ 9/17	2/ 5/21	3/ 2/23	3/ 4/21
b) $D = 20$					
FEs	vs $K = 1$ (w/o PV)	vs $K = 2$	vs $K = 4$	vs $K = 8$	vs $K = 10$
2000	0/16/12	0/15/13	0/ 1/27	2/ 0/26	3/ 0/25
4000	0/18/10	0/13/15	0/ 3/25	3/ 0/25	4/ 0/24
6000	0/13/15	1/13/14	0/ 1/27	3/ 1/24	3/ 2/23
8000	1/11/16	1/ 8/19	1/ 1/26	2/ 2/24	1/ 2/25
10,000	1/ 8/19	1/ 7/20	2/ 1/25	2/ 2/24	1/ 5/22
c) $D = 30$					
FEs	vs $K = 1$ (w/o PV)	vs $K = 2$	vs $K = 4$	vs $K = 8$	vs $K = 10$
2000	1/17/10	0/10/18	1/ 3/24	2/ 0/26	5/ 0/23
4000	1/15/12	0/ 9/19	0/ 0/28	2/ 1/25	3/ 0/25
6000	0/14/14	0/11/17	0/ 0/28	1/ 2/25	1/ 1/26
8000	0/ 9/19	0/ 5/23	0/ 0/28	1/ 2/25	1/ 2/25
10,000	3/ 7/18	1/ 5/22	0/ 0/28	0/ 2/26	2/ 3/23

Table 11 presents the statistical results in terms of the count of $+/-/\sim$. The symbols are defined as in the previous subsection, where the default value of M is $M = 25$. To highlight the variant with individual-based adaptation, "(Indiv.)" is appended to $M = 100$ in the table. When individual-based adaptation was conducted ($M = 100$), the performance of EBADE degraded significantly on multiple problem instances compared with EBADE using the default value $M = 25$, confirming the effectiveness of the proposed subpopulation-based adaptation. This is because the proposed subpopulation-based adaptation can improve the validation accuracy of the effectiveness of parameter configuration by validating those with multiple samples. Specifically, the subpopulation-based adaptation can mitigate the risk of a sample moving in an unintended direction due to random numbers, resulting in an unjustified evaluation of parameter configurations. However, the individual-based adaptation is designed to validate a parameter configuration using a single sample, and thereby, this risk should occur frequently. Accordingly, EBADE performed well using good parameter configurations, which were well detected by the subpopulation-based adaptation.

Moreover, EBADE with $M \in \{2, 4, 5\}$ should be avoided as they statistically underperformed compared to EBADE with the default setting ($M = 25$). These results suggest that the number of parameter configurations tested in parallel should be greater than five and the number of solutions validating one parameter configuration should be less than 20. Further, EBADE with $M = 50$ is also inappropriate, because two solutions are insufficient to validate one parameter configuration. Increasing the value of N while decreasing M can improve this validation accuracy, because the number of validation samples, N, increases; however, the number of parameter configurations decreases, leading to a poor diversity of solutions. On the other hand, increasing M (and thus decreasing N) can improve the diversity of solutions; however, less effective configurations may be used owing to low validation accuracy. Thus, it is important to use a parameter setting of $\{N, M\}$ that balances this trade-off. Our experimental result suggests that $\{N, M\} = \{10, 10\}, \{5, 20\}, \{4, 25\}$ produces a good balance, as EBADE with these settings performed well.

Adaptation results

Finally, we verify whether the high performance of EBADE was achieved after EBADE performed adaptation without biasing the use of only certain DE parameter configurations. Specifically, for each element of the parameter configurations, i.e., the scaling factor F, the crossover rate CR, the mutation strategy, and the crossover strategy, we enumerated the number of times the subpopulations of EBADE used each

Table 11 Statistical results (the count of $+/-/\sim$) for 2000, 4000, 6000, 8000, and 10,000 FEs in comparison with EBADE with $M \in \{2, 4, 5, 10, 20, 50, 100\}$

a) $D = 10$							
FEs	vs $M = 2$	vs $M = 4$	vs $M = 5$	vs $M = 10$	vs $M = 20$	vs $M = 50$	vs $M = 100$ (Indiv.)
2000	0/ 7/21	0/ 6/22	0/ 4/24	2/ 1/25	0/ 1/27	1/ 1/26	0/ 0/28
4000	1/12/15	1/ 8/19	0/ 5/23	1/ 1/26	0/ 1/27	0/ 3/25	1/ 3/24
6000	0/15/13	0/10/18	0/ 7/21	0/ 1/27	0/ 1/27	0/ 3/25	1/ 4/23
8000	0/12/16	0/ 7/21	0/ 7/21	0/ 0/28	0/ 0/28	1/ 4/23	0/ 6/22
10,000	2/11/15	0/ 8/20	0/ 6/22	0/ 0/28	0/ 1/27	0/ 5/23	0/ 6/22
b) $D = 20$							
FEs	vs $M = 2$	vs $M = 4$	vs $M = 5$	vs $M = 10$	vs $M = 20$	vs $M = 50$	vs $M = 100$ (Indiv.)
2000	0/10/18	0/ 1/27	0/ 4/24	0/ 0/28	0/ 0/28	1/ 4/23	0/ 1/27
4000	0/13/15	0/ 2/26	1/ 4/23	0/ 1/27	1/ 1/26	0/ 1/27	1/ 1/26
6000	0/ 9/19	0/ 2/26	1/ 3/24	2/ 1/25	2/ 1/25	0/ 2/26	1/ 3/24
8000	0/ 8/20	1/ 5/22	1/ 3/24	1/ 0/27	2/ 0/26	0/ 2/26	0/ 5/23
10,000	1/ 6/21	1/ 7/20	1/ 4/23	2/ 1/25	1/ 1/26	0/ 2/26	0/ 4/24
c) $D = 30$							
FEs	vs $M = 2$	vs $M = 4$	vs $M = 5$	vs $M = 10$	vs $M = 20$	vs $M = 50$	vs $M = 100$ (Indiv.)
2000	0/ 8/20	0/ 5/23	0/ 3/25	3/ 1/24	1/ 1/26	0/ 1/27	2/ 2/24
4000	0/ 9/19	0/ 2/26	0/ 2/26	0/ 1/27	1/ 1/26	0/ 1/27	1/ 1/26
6000	0/10/18	0/ 4/24	0/ 4/24	0/ 1/27	0/ 1/27	0/ 3/25	1/ 3/24
8000	1/12/15	1/ 4/23	1/ 4/23	0/ 1/27	0/ 1/27	0/ 2/26	1/ 6/21
10,000	1/10/17	1/ 4/23	2/ 4/22	0/ 1/27	0/ 0/28	0/ 2/26	1/ 5/22

candidate within 10,000 FEs, and reported it as a ratio. For F and CR, the domains of definition $F, CR \in [0, 1]$ were divided into five ranges with widths of 0.2, and the number of samples was examined for each range.

Figure 3 summarizes the ratio of each candidate used by problem function and the dimension. Because the selected ratio did not depend on D significantly, the results are reported with $D \in \{10, 30\}$ and those with $D = 20$ are omitted. As illustrated in this figure, EBADE selected various parameter configurations of F, CR, and the mutation and crossover strategies over the search process. As in Fig. 3a), values of the scaling factor F in $[0.0, 0.2]$ were most frequently used, while higher values like $F \in (0.8, 1.0]$ were less frequently used. This may be attributed to premature convergence to the local optima induced by larger values or slow convergence as the solutions continue to move widely through the search space. On the other hand, larger crossover rate values CR were often selected, as illustrated in Fig. 3b), especially in unimodal functions, i.e., F1–F5. In other words, mutant solutions generated with mutation strategies to accelerate convergence should be actively utilized in EOPs. The *best/1* mutation strategy was the most selected, as depicted in Fig. 3c). This tendency is natural, since the *best/1* strategy exhibits the strongest exploitation ability, which is suitable for EOPs. However, the other strategies were used adequately to maintain the diversity of solutions and the combined use of these four strategies contributed to the high performance of EBADE, which can be also confirmed in the comparison between EBADE and DE with *best/1/bin* or *best/1/exp* in Sect. 5.1. In Fig. 3 d), the *binomial* crossover strategy was used slightly more. This tendency was reinforced as D was increased. Since the importance of solution diversity became more notable as D was increased, the *binomial* strategy, which conducts crossover more uniformly, was more suitable. In summary, EBADE exhibited high performance by selecting more candidates appropriate for EOPs. However, it avoided heavy bias by adaptively selecting all of them.

Conclusions

In this paper, we introduced a new adaptive DE variant named emulation-based adaptive DE (EBADE). An adaptive EA for EOPs with moderately restricted budgets was developed for the first time by emulating the principle of sample-efficient approaches, e.g., SAEAs. Specifically, EBADE is characterized by prior validation and subpopulation-based adaptation; emulating the EI-based solution screening with surrogates

Fig. 3 Adaptation results obtained from EBADE

in SAEAs. EBADE pre-screens for expected improvements from candidate DE parameter configurations before using them without any surrogate. It adopts a multi-population mechanism and each parameter configuration is assigned to a subpopulation to determine the effectiveness of parameter configurations accurately by validating one configuration with respect to multiple solutions. The experimental results obtained in moderately expensive cases demonstrated the statistically significant superiority of EBADE among popular and modern adaptive DEs. Additionally, EBADE was also highly competitive with state-of-the-art SAEAs in moderately EOPs; while SAEAs induced premature convergence, EBADE continued to improve in performance with the shortest runtime. Consequently, this paper contributes to solving

moderately EOPs with high-performance, computationally efficient, and auto-tunable approaches.

Despite the effectiveness of EBADE, some possible drawbacks can be pointed out. First, EBADE employs FIR to detect good parameter configurations, but this may be hindered until EBADE discovers good solutions. Accordingly, we will explore alternative indicators instead of FIR to detect good parameter configurations even with a small improvement in fitness values. Second, the Euclidean distance is utilized in the prior validation phase, but it may not be adequate when the problem dimension increases. Thus, we will consider other metrics such as the Mahalanobis distance and cosine similarity to improve the scalability of EBADE to problem dimensions. Finally, we plan to extend this framework for multi-objective optimization problems and compare the performances of the extended EBADE and multi-objective SAEAs.

Acknowledgements This work was supported by JSPS KAKENHI under Grant Nos. 22J21254 and 20H04254.

Data Availability The source code that supports the findings of this study is openly available at https://github.com/YNU-NakataLab/EBADE.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Baskar S, Miruna JAS (2015) Surrogate assisted-hybrid differential evolution algorithm using diversity control. Expert Syst 4(32):531–545. https://doi.org/10.1111/exsy.12105
2. Baumert T, Brixner T, Seyfried V et al (1997) Femtosecond pulse shaping by an evolutionary algorithm with feedback. Appl Phys B 65(6):779–782. https://doi.org/10.1007/s003400050346
3. Brest J, Greiner S, Boskovic B et al (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657. https://doi.org/10.1109/TEVC.2006.872133
4. Brest J, Maučec MS, Bošković B (2017) Single objective real-parameter optimization: Algorithm jSO. In: IEEE Congr. Evol. Comput. (CEC), pp 1311–1318. https://doi.org/10.1109/CEC.2017.7969456
5. Briffoteaux G (2022) Parallel surrogate-based algorithms for solving expensive optimization problems. PhD thesis, University of Lille
6. Cai Y, Wang J (2013) Differential evolution with neighborhood and direction information for numerical optimization. IEEE Trans Cybern 43(6):2202–2215. https://doi.org/10.1109/TCYB.2013.2245501
7. Cai X, Gao L, Li X et al (2019) Surrogate-guided differential evolution algorithm for high dimensional expensive problems. Swarm Evol Comput 48:288–311. https://doi.org/10.1016/j.swevo.2019.04.009
8. Cheng R, He C, Jin Y et al (2018) Model-based evolutionary algorithms: a short survey. Complex Intell Syst 4(4):283–292. https://doi.org/10.1007/s40747-018-0080-1
9. Chowdhury R, Adhikari S (2011) Reliability analysis of uncertain dynamical systems using correlated function expansion. Int J Mech Sci 53(4):281–285. https://doi.org/10.1016/j.ijmecsci.2011.01.009
10. Chugh T, Sindhya K, Hakanen J et al (2019) A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Comput 23(9):3137–3166. https://doi.org/10.1007/s00500-017-2965-0
11. Elsayed SM, Ray T, Sarker RA (2014) A surrogate-assisted differential evolution algorithm with dynamic parameters selection for solving expensive optimization problems. In: IEEE Congr. Evol. Comput. (CEC), pp 1062–1068. https://doi.org/10.1109/CEC.2014.6900351
12. Hajela P (1990) Genetic search—an approach to the nonconvex optimization problem. AIAA J 28(7):1205–1210. https://doi.org/10.2514/3.25195
13. He C, Zhang Y, Gong D et al (2023) A review of surrogate-assisted evolutionary algorithms for expensive optimization problems. Expert Syst Appl 217:119495. https://doi.org/10.1016/j.eswa.2022.119495
14. Islam SM, Das S, Ghosh S et al (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans Syst Man Cybern B Cybern 42(2):482–500. https://doi.org/10.1109/TSMCB.2011.2167966
15. Jia L, Gong W, Wu H (2009) An improved self-adaptive control parameter of differential evolution for global optimization. In: Comput. Intell. Intell. Syst. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 215–224. https://doi.org/10.1007/978-3-642-04962-0_25
16. Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 1(2):61–70. https://doi.org/10.1016/j.swevo.2011.05.001
17. Jin Y, Olhofer M, Sendhoff B (2001) Managing approximate models in evolutionary aerodynamic design optimization. In: IEEE Congr. Evol. Comput. (CEC), pp 592–599. https://doi.org/10.1109/CEC.2001.934445
18. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Global Optimiz 13(4):455–492. https://doi.org/10.1023/a:1008306431147
19. Karafotias G, Hoogendoorn M, Eiben AE (2015) Parameter control in evolutionary algorithms: trends and challenges. IEEE Trans Evol Comput 19(2):167–187. https://doi.org/10.1109/TEVC.2014.2308294
20. Kramer O (2010) Evolutionary self-adaptation: a survey of operators and strategy parameters. Evol Intell 3(2):51–65. https://doi.org/10.1007/s12065-010-0035-y
21. Krempser E, Bernardino HS, Barbosa HJC, et al (2012) Differential evolution assisted by surrogate models for structural optimization

problems. In: Int. Conf. Eng. Comput. Technol., vol 49. Civil-Comp Press, Stirlingshire, UK. https://doi.org/10.4203/ccp.100.49
22. Li G, Lin Q, Cui L et al (2016) A novel hybrid differential evolution algorithm with modified CoDE and JADE. Appl Soft Comput 47:577–599. https://doi.org/10.1016/j.asoc.2016.06.011
23. Li Y, Han T, Zhou H et al (2022) A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem. Inf Sci 606:350–367. https://doi.org/10.1016/j.ins.2022.05.058
24. Li G, Xie L, Wang Z et al (2023) Evolutionary algorithm with individual-distribution search strategy and regression-classification surrogates for expensive optimization. Inf Sci 634:423–442. https://doi.org/10.1016/j.ins.2023.03.101
25. Liang JJ, Qu BY, Suganthan PN, et al (2013) Problem eefinitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. In: Tech. rep., Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, and Nanyang Tech. Univ., Singapore
26. Liu B, Zhang Q, Gielen GGE (2014) A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. IEEE Trans Evol Comput 18(2):180–192. https://doi.org/10.1109/TEVC.2013.2248012
27. Lobo FJ, Lima CF, Michalewicz Z (2007) Parameter setting in evolutionary algorithms. Springer, Berlin, Heidelberg
28. Lu XF, Tang K (2012) Classification- and regression-assisted differential evolution for computationally expensive problems. J Comput Sci Technol 27(5):1024–1034. https://doi.org/10.1007/s11390-012-1282-4
29. Lu X, Tang K, Sendhoff B et al (2014) A new self-adaptation scheme for differential evolution. Neurocomputing 146:2–16. https://doi.org/10.1016/j.neucom.2014.04.071
30. Lu X, Tang K, Yao X (2011) Classification-assisted differential evolution for computationally expensive problems. In: IEEE Congr. Evol. Comput. (CEC), pp 1986–1993. https://doi.org/10.1109/CEC.2011.5949859
31. Lynn N, Mallipeddi R, Suganthan PN (2015) Differential evolution with two subpopulations. In: Swarm Evol. Memet. Comput. Springer International Publishing, New York, NY, USA, pp 1–13. https://doi.org/10.1007/978-3-319-20294-5_1
32. Ma X, Zhang K, Zhang L et al (2022) A distributed surrogate system assisted differential evolutionary algorithm for computationally expensive history matching problems. J Pet Sci Eng 210:110029. https://doi.org/10.1016/j.petrol.2021.110029
33. Mallipeddi R, Lee M (2012) Surrogate model assisted ensemble differential evolution algorithm. In: IEEE Congr. Evol. Comput. (CEC), pp 1–8. https://doi.org/10.1109/CEC.2012.6256479
34. Mallipeddi R, Lee M (2015) An evolving surrogate model-based differential evolution algorithm. Appl Soft Comput 34:770–787. https://doi.org/10.1016/j.asoc.2015.06.010
35. Mallipeddi R, Suganthan PN, Pan QK et al (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl Soft Comput 11(2):1679–1696. https://doi.org/10.1016/j.asoc.2010.04.024
36. Nair P, Keane A, Shimpi R (1998) Combining approximation concepts with genetic algorithm-based structural optimization procedures. In: AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dynamics Mater. Conf. Exhib. American Institute of Aeronautics and Astronautics, Reston, Virigina. https://doi.org/10.2514/6.1998-1912
37. Nishihara K, Nakata M (2021) Performance improvement with prior-validation framework for algorithmic configuration on self-adaptive differential evolution. Trans Math Model Appl 14(3):51–67 (in Japanese)
38. Nishihara K, Nakata M (2022) Surrogate-assisted differential evolution with adaptation of training data selection criterion. In: IEEE Symp. Ser. Comput. Intell. (SSCI), pp 1675–1682. https://doi.org/10.1109/SSCI51031.2022.10022105
39. Oyama A, Kohira T, Kemmotsu H, et al (2017) Simultaneous structure design optimization of multiple car models using the K computer. In: IEEE Symp. Ser. Comput. Intell. (SSCI), pp 1–4. https://doi.org/10.1109/SSCI.2017.8285350
40. Pan JS, Liu N, Chu SC et al (2021) An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. Inf Sci 561:304–325. https://doi.org/10.1016/j.ins.2020.11.056
41. Piotrowski AP (2018) L-SHADE optimization algorithms with population-wide inertia. Inf Sci 468:117–141. https://doi.org/10.1016/j.ins.2018.08.030
42. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417. https://doi.org/10.1109/TEVC.2008.927706
43. Rao RV, Patel V (2013) An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. Sci Iran 20(3):710–720. https://doi.org/10.1016/j.scient.2012.12.005
44. Shan S, Wang GG (2010) Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct Multidiscip Optim 41(2):219–241. https://doi.org/10.1007/s00158-009-0420-2
45. Sharma M, Komninos A, López-Ibáñez M, et al (2019) Deep reinforcement learning based parameter control in differential evolution. In: Annu. Conf. Genet. Evol. Comput. (GECCO). ACM, New York, NY, USA, GECCO '19, pp 709–717. https://doi.org/10.1145/3321707.3321813
46. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optimiz 11:341–359. https://doi.org/10.1023/A:1008202821328
47. Sun C, Jin Y, Zeng J et al (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. Soft Comput 19(6):1461–1475. https://doi.org/10.1007/s00500-014-1283-z
48. Sun G, Yang B, Yang Z et al (2020) An adaptive differential evolution with combined strategy for global numerical optimization. Soft Comput 24(9):6277–6296. https://doi.org/10.1007/s00500-019-03934-3
49. Tan Z, Li K, Wang Y (2021) Differential evolution with adaptive mutation strategy based on fitness landscape analysis. Inf Sci 549:142–163. https://doi.org/10.1016/j.ins.2020.11.023
50. Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for Differential Evolution. In: IEEE Congr. Evol. Comput. (CEC), pp 71–78. https://doi.org/10.1109/CEC.2013.6557555
51. Tanabe R, Fukunaga AS (2014) Improving the search performance of SHADE using linear population size reduction. In: IEEE Congr. Evol. Comput. (CEC), pp 1658–1665. https://doi.org/10.1109/CEC.2014.6900380
52. Tanabe R, Fukunaga A (2020) Reviewing and benchmarking parameter control methods in differential evolution. IEEE Trans Cybern 50(3):1170–1184. https://doi.org/10.1109/TCYB.2019.2892735
53. Tirronen V, Neri F (2009) Differential Evolution with Fitness Diversity Self-adaptation. In: Chiong R (ed) Nature-inspired algorithms for optimisation. Springer, Berlin Heidelberg, Berlin, Heidelberg, p 199–234. https://doi.org/10.1007/978-3-642-00267-0_7
54. Tolson BA, Shoemaker CA (2007) Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. Water Resour Res 43(1):1–16. https://doi.org/10.1029/2005WR004723
55. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters.

IEEE Trans Evol Comput 15(1):55–66. https://doi.org/10.1109/TEVC.2010.2087271
56. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. IEEE Trans Cybern 47(9):2664–2677. https://doi.org/10.1109/TCYB.2017.2710978
57. Wang X, Wang GG, Song B et al (2019) A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. IEEE Trans Evol Comput 23(5):815–827. https://doi.org/10.1109/TEVC.2019.2890818
58. Wu G, Mallipeddi R, Suganthan PN et al (2016) Differential evolution with multi-population based ensemble of mutation strategies. Inf Sci 329:329–345. https://doi.org/10.1016/j.ins.2015.09.009
59. Wu G, Shen X, Li H et al (2018) Ensemble of differential evolution variants. Inf Sci 423:172–186. https://doi.org/10.1016/j.ins.2017.09.053
60. Yu H, Tan Y, Zeng J et al (2018) Surrogate-assisted hierarchical particle swarm optimization. Inf Sci 454–455:59–72. https://doi.org/10.1016/j.ins.2018.04.062
61. Zhan ZH, Zhang J, Li Y et al (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern B Cybern 39(6):1362–1381. https://doi.org/10.1109/TSMCB.2009.2015956
62. Zhang J, Sanderson AC (2007) DE-AEC: a differential evolution algorithm based on adaptive evolution control. In: IEEE Congr. Evol. Comput. (CEC), pp 3824–3830. https://doi.org/10.1109/CEC.2007.4424969
63. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958. https://doi.org/10.1109/TEVC.2009.2014613
64. Zhu H, Zhang H, Jin Y (2021) From federated learning to federated neural architecture search: a survey. Complex Intell Syst 7(2):639–657. https://doi.org/10.1007/s40747-020-00247-z