Adaptive dropout for high-dimensional expensive multiobjective optimization

Jianqing Lin¹ · Cheng He¹ · Ran Cheng¹

Received: 2 March 2021 / Accepted: 29 March 2021 / Published online: 21 April 2021
© The Author(s) 2021

Abstract

Various works have been proposed to solve expensive multiobjective optimization problems (EMOPs) using surrogate-assisted evolutionary algorithms (SAEAs) in recent decades. However, most existing methods focus on EMOPs with less than 30 decision variables, since a large number of training samples are required to build an accurate surrogate model for high-dimensional EMOPs, which is unrealistic for expensive multiobjective optimization. To address this issue, we propose an SAEA with an adaptive dropout mechanism. Specifically, this mechanism takes advantage of the statistical differences between different solution sets in the decision space to guide the selection of some crucial decision variables. A new infill criterion is then proposed to optimize the selected decision variables with the assistance of surrogate models. Moreover, the optimized decision variables are extended to new full-length solutions, and then the new candidate solutions are evaluated using expensive functions to update the archive. The proposed algorithm is tested on different benchmark problems with up to 200 decision variables compared to some state-of-the-art SAEAs. The experimental results have demonstrated the promising performance and computational efficiency of the proposed algorithm in high-dimensional expensive multiobjective optimization.

Keywords Surrogate-assisted optimization · Evolutionary algorithm · High-dimensional · Multiobjective optimization · Adaptive dropout

Introduction

Multiobjective optimization problems (MOPs) refer to the optimization of multiple conflicting objectives simultaneously, i.e., improvement in one objective may lead to the degeneration of at least one of the other objectives [1]. MOPs are commonly seen in the science and engineering areas [2], e.g., neural architecture search [3], structure optimization [4], and industrial dispatch [5]. An MOP can be mathematically formulated as

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ &\text{subject to } \mathbf{x} \in X, \end{aligned} \quad (1)$$

where X is the search space of decision variables, $m (\geq 2)$ is the number of objectives, and $\mathbf{x} = (x_1, \dots, x_D)$ is the decision vector with D decision variables [6].

The goal of multiobjective optimization is to obtain a set of representative Pareto-optimal solutions [7]. To be more specific, each solution in the Pareto-optimal solution set (PS) is not Pareto dominated by other solutions in X . Notably, the Pareto dominance between solution $\mathbf{x}_1$ and solution $\mathbf{x}_2$ is notated as $\mathbf{x}_1 \prec \mathbf{x}_2$ if and only if $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ for all $1 \leq i \leq m$, and $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ for at least one $1 \leq j \leq m$ [8].

As a complex case of MOPs, if the evaluation of any objective function in an MOP is computationally or economically expensive, the problem is called an expensive multiobjective optimization problem (EMOP) [9]. For example, in an automobile body structural design problem, one structural simulation takes tens of minutes or even hours [10]. As a powerful tool for black-box optimization, the evolutionary algorithms (EAs) are proven to be helpful in solving MOPs.

✉ Ran Cheng
ranchengcn@gmail.com

Jianqing Lin
linjqcn@gmail.com

Cheng He
chenghehust@gmail.com

¹ Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Nevertheless, EAs are not competent to solve EMOPs, since the requirement of massive function evaluations leads to unbearable time consumption and computational cost [11].

Surrogate-assisted EAs (SAEAs), which apply surrogate model techniques to traditional evolutionary algorithms, are the mainstream methods to solve EMOPs. Surrogate models are used to replace part of computationally expensive function evaluations (FEs). Since the surrogate models are computationally cheap and efficient, they can significantly reduce the required computational costs [12]. The most commonly used surrogate models include polynomial regression (PR) [13], Gaussian process (GP) [13,14], radial basis function (RBF) [15,16], artificial neural networks (ANN) [11], etc. Apart from the adopted surrogate model(s), model management is also a crucial part of SAEAs, which mainly involves how to update the current surrogate model in two aspects. First, it includes the selection of offspring individuals to be re-evaluated by expensive function(s). Second, the infill criterion determines the quality of the candidate re-evaluated individuals, where the re-evaluated solutions will be added to the training dataset for updating the surrogate model [17].

In the past decades, a number of SAEAs have been proposed to solve EMOPs effectively. The most representative SAEAs include ParEGO [18], MOEA/D-EGO [19], K-RVEA [20,21], CSEA [11], etc. However, the SAEAs above only focus on low-dimensional optimization with less than 30 decision variables [18,19,21,22]. It is mainly due to the lack of training samples for building accurate surrogate models and the unbearable time cost for training the models with high-dimensional input. Recently, several attempts are made to address this issue. Some methods use dimensionality reduction techniques to map decision variables from high-dimensional space to low-dimensional space and then optimize in low-dimensional space [23–25]. Other approaches try to use other cheap surrogate models instead of computationally expensive Gaussian process models, or use some multiswarm optimization algorithms rather than conventional genetic algorithms [26]. Very recently, EDN-ARMOEA, a new method was proposed for solving high-dimensional EMOPs, which used a dropout neural network instead of the Gaussian process [27]. Nevertheless, the performance of the SAEAs mentioned above can be further improved.

In this paper, a surrogate-assisted particle swarm optimization algorithm with an adaptive dropout mechanism, called ADSAPSO, is proposed to handle high-dimensional EMOPs. We mainly focus on selecting a small number of crucial decision variables for convergence enhancement and diversity maintenance, aiming to reduce the cost for building and training surrogate model(s). Compared with EDN-ARMOEA, which uses a dropout ANN for dimension

reduction, the proposed ADSAPSO is different mainly in two aspects. (i) EDN-ARMOEA uses the dropout strategy for improving the computational efficiency of the surrogate model; by contrast, the dropout strategy in ADSAPSO is adopted to reduce the dimension of the optimization problem. (ii) The dropout neural network in EDN-ARMOEA discards neurons of the network randomly with a certain probability. On the contrary, the dropout mechanism in ADSAPSO learns the deterministic reduction via a statistic model. The main new contributions of this work are summarized as follows:

- (1) Inspired by the dropout strategy in deep learning, we proposed an adaptive dropout mechanism to select several decision variables for dimensionality reduction. Unlike existing unsupervised dimensionality reduction approaches, the proposed adaptive mechanism learns from the statistical differences of different solution sets in the decision space for better dimension reduction.
- (2) To select promising candidate solutions to be re-evaluated, a novel infill criterion is proposed, which contains the RBF-based search and the Replacement operation. Notably, We use the RBF-based search to optimize the selected d decision variables, and the Replacement operation is used to extend the low-dimensional decision vectors to full-length candidate solutions.

The rest of this paper is organized as follows. In “Background”, we provide the background of the work, including the dropout mechanism, the RBF model, and the estimation of distribution model. The details of the proposed algorithm are illustrated in “Proposed method”. The experimental studies and result analysis are given in “Experimental study”. Finally, “Conclusion” concludes this paper.

Background

In this section, we first summarize the dropout in deep learning, which is extended to dimension dropout in this study. Next, the detailed principle and components of the RBF are illustrated. Then we introduce the estimation of distribution model, which is used in our proposed adaptive dropout mechanism.

Dropout

Dropout, which was first proposed in Ref. [28], originally referred to temporarily discarding neurons from the network according to a certain probability during the forward propagation process of the deep neural network, which is equivalent to finding a thinner network from the original net-

Fig. 1 An example of the dropout in neural network. **a** represents the original Neural Network. **b** Shows the Neural Network after dropout

work. An example of the dropout in a neural network is shown in Fig. 1.

The original neural network is shown in Fig. 1a. After dropout, some neurons are discarded, as shown by the dotted circle in Fig. 1b. The complexity of the neural network is significantly reduced, and it can make the neural network more general, which reduces the reliance on some local features and effectively alleviates the occurrence of over-fitting.

More specifically, for a neural network with N neurons, it can be regarded as a collection of 2^N neural networks after dropout, which can prevent over-fitting while improving the training efficiency of networks.

Radial basis function (RBF) neural network

The RBF neural network is a simple and widely used surrogate model that approximates any nonlinear function. The RBF neural network has a fast training speed due to that it is a local approximation network. On the basis of training data $\{(\mathbf{x}_i, y_i) | i = 1, \dots, N\}$, it approximates a continuous function as follows:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\mathbf{x}_i, \mathbf{x}), \quad (2)$$

where w_i is the weight coefficient, $\phi(\cdot, \cdot)$ represents the kernel function and the distance between the point $\mathbf{x}_i$ and the point $\mathbf{x}$ is used as the independent variable of the kernel function.

Commonly used kernel functions include linear kernel function, polynomial kernel function, Gaussian kernel function, etc. For example, the Gaussian kernel function is calculated as follows:

$$\phi(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\delta^2}\right), \quad (3)$$

where δ is the expansion constant of kernel function, which reflects the width of the kernel function image. The smaller the δ , the narrower the width.

The weight vector $\mathbf{w} = (w_1, \dots, w_N)$ is calculated as follows:

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}, \quad (4)$$

where $\mathbf{y} = (y_1, \dots, y_N)$ is the output vector of the training data and Φ is the matrix computed as follows:

$$\Phi = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N, \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}. \quad (5)$$

Statistical model

Statistical model refers to a model built by mathematical-statistical methods, which obtains the relationship and the development trend between variables through mathematical statistics [29].

Mathematical statistics has two functions: descriptive statistics and inferential statistics. Central trend statistics is a standard descriptive statistical method, which uses a trend value to represent the overall level of a set of data. Commonly used trend values include average, median, and mode. Discrete trend analysis reflects the degree of dispersion of data, and its commonly used indicators are range and standard deviation. Frequency analysis mainly describes the number of occurrences of each characteristic value of the data, reflecting the distribution of a set of data.

Inferential statistics is divided into two parts: parameter estimation and hypothesis testing. The former is based on a one-time sampling experiment to estimate a certain characteristic of the entire data set. The latter test a certain hypothesis and infer whether the hypothesis made is acceptable according to the calculation results.

Proposed method

In this section, the main framework of the proposed ADSAPSO is introduced, followed by the details of the proposed model management, including the adaptive dropout mechanism and a new infill criterion.

Framework of ADSAPSO

The framework of the proposed ADSAPSO is presented in Algorithm 1, which mainly consists of the adaptive dropout mechanism and the infill criterion. First, N solutions are sampled using the Latin hypercube sampling (LHS) strategy [30]

and evaluated using the expensive functions. Then these evaluated candidate solutions are merged into an empty archive Arc. Next, the environmental selection selects N_α solutions (arc) from archive Arc. Afterwards, the adaptive dropout mechanism and the infill criterion are adopted. The adaptive dropout mechanism aims to select the d -dimensional variables, and the infill criterion is used to choose the new samples to be evaluated by expensive functions. Notably, we first select one well-performing solution set and one poorly-performing solution set, which are statistically different in the decision space. Accordingly, d -dimensional variable set S is selected from the full-length decision variables, where the chosen variables significantly affect the convergence. As for the infill criterion, the RBF-based search is used to optimize the selected d -dimensional decision variables and obtain optimal k d -dimensional decision vector set X_d . Then a replacement operator selects k full-length solutions, whose corresponding decision variables will be replaced by X_d , to form individual set X_{new} . X_{new} will be evaluated using expensive functions for updating the archives Arc. Finally, the obtained Arc will be output as the final solutions. Note that ADSAPSO uses the same environmental selection operator in NSGA-II [7], and we will not describe its details.

Algorithm 1 The Framework of the ADSAPSO

Input: N (number of initial solutions), $\mathbf{f}$ (expensive functions), MaxFES (total FES), N_α (number of samples for building the surrogate model), β (probability of Dimension Dropout), k (number of re-evaluated individuals).

Output: Evaluated solutions in Arc

```

1: Initialize:
2:  $X_P, Y_P \leftarrow LHS(N)$  based initialization
3:  $\text{Arc} \leftarrow \{X_P, Y_P\}$ 
4:  $\text{FES} \leftarrow N$ 
5: while  $\text{FES} < \text{MaxFES}$  do
6:    $\text{arc} \leftarrow \text{Environmental\_Selection}(\text{Arc}, N_\alpha)$ 
7:   /**** Adaptive Dropout Mechanism ****/
8:    $S \leftarrow \text{Dimension\_Dropout}(\text{arc}, \beta)$ 
9:    $\text{arc}_d \leftarrow \{X_P \setminus S, Y_P\}$ 
10:  /***** Infill Criterion *****/
11:   $X_d \leftarrow \text{RBF-based\_Search}(\text{arc}_d, k)$ 
12:   $X_{\text{new}} \leftarrow \text{Replacement}(X_d, \text{arc}, k)$ 
13:   $Z_{\text{new}} \leftarrow \{X_{\text{new}}, \mathbf{f}(X_{\text{new}})\}$ 
14:   $\text{Arc} \leftarrow \text{Arc} \cup Z_{\text{new}}$ 
15:   $\text{FES} \leftarrow \text{FES} + k$ 
16: end while
  
```

Fig. 2 An illustrative example of the proposed adaptive dropout mechanism, which reduces the dimension from D to d

Adaptive dropout mechanism

The adaptive dropout mechanism is a crucial component of the proposed ADSAPSO (line 8 in Algorithm 1), which discards the decision dimensions from D dimensions to d . In D -dimensional decision variables, some variables have a more significant effect on convergence enhancement, and they are important for expensive multiobjective optimization. Thus, the adaptive dropout mechanism adapts a statistical model-assisted method for selecting d decision variables that significantly affect convergence from the original full-length decision variables. An example of the adaptive dropout mechanism is shown in Fig. 2, where the dashed circle indicates the discarded decision variables.

Unlike the principal component analysis technique [31], which maps data from high-dimensional space to low-dimensional space, our proposed adaptive dropout mechanism retains the original information of the selected d dimensions. Specifically, the proposed adaptive dropout mechanism learns from different solution sets' statistical differences in the decision space.

Generally, the proposed adaptive dropout mechanism consists of three steps: (1) Solutions selection; (2) Statistical models construction; (3) Dimension selection.

(1) *Solutions selection:* Assuming that there are significant differences in some dimensions in the decision space between a set of well-performing solutions and a set of poorly-performing ones. The distribution differences of these two solution sets can somehow reflect the importance of some decision variables in convergence enhancement. Thus, a suitable solution selection should be used to capture the differences. If well-performing solutions and poorly-performing ones are selected from archive Arc, the poorly-performing solutions remain the same during the update process. In this case, the rankings of poorly-performing solutions in the whole Arc are almost unchanged. In other words, the distribution of the poorly-performing solution sets will not help in the later stages of evolution. To remedy this issue, we select the well-performing solutions and the poorly-performing solutions from archive arc, where solutions in arc are elite solutions selected from Arc. Afterwards, according to the non-dominated sorting, the first N_s solutions in arc

are regarded as well-performing solutions, and the last N_s solutions in *arc* are deemed to be poorly-performing ones. Consequently, the indistinguishable situation can be avoided.

(2) *Statistical models construction* After solutions selection, two statistical models are constructed using the selected well-performing solutions and the selected poorly-performing solutions, respectively. To discover the effect of different variables, we analyze the two sets of solutions in the decision space via the constructed two statistical models. To be more specific, we count the average value of each dimension of the two solution sets as the statistical value separately. Since the selected two sets of solutions have significant differences in the objective space, the influence of different variables on the objective values can be reflected by the statistical differences in the decision space. Thus, dimensions with significant differences in the decision space will be emphasized for better convergence enhancement in the following optimization.

(3) *Dimension selection* After constructing two statistical models, a difference model is obtained by calculating the difference between each dimension in the two models. Then, we select the top $\beta \cdot D$ dimensions with the highest absolute difference to set S .

Figure 3 shows an example of the adaptive dropout mechanism, and Fig. 3a demonstrates the detailed process of solutions selection. In this figure, the blue squares are the “Good” solutions and the red triangles are the “Bad” ones. Notably, the black circles on the upper right corner in Fig. 3a are the solutions (in *Arc*) that are not selected by *arc*. They will not be selected as “Bad” solutions for preventing the distribution of “Bad” solutions from premature. Figure 3b presents two statistical models, where the horizontal axis represents the dimension, and the vertical one is the dimension value. Besides, the red line indicates the model of poorly-performing solutions (“Bad” solutions), and the blue line indicates the model of well-performing solutions (“Good” solutions). Figure 3c demonstrates the absolute difference of the above two statistical models, where the horizontal axis represents the dimension, and the vertical axis represents the absolute difference between two statistical models. The dotted line represents the selection threshold, and the shaded parts indicate the dimensions to be selected. Specifically, d_1 , d_2 , d_3 , and d_4 represent the lower and upper boundaries that locating to the dimensions above the threshold. The corresponding dimensions in $[d_1, d_2]$ and $[d_3, d_4]$ are the dimensions we ultimately choose.

For high-dimensional EMOPs, a large number of training samples are required for constructing accurate surrogate models, which is unrealistic for expensive multiobjective optimization. Our proposed adaptive dropout mechanism reduces the dimension of decision variables from D to d according to the statistical results. Generally, the proposed method helps build relatively accurate d -dimensional surrogate models with the same number of training samples.

Fig. 3 Illustrative example of adaptive dropout mechanism. **a** Shows the distribution of the solutions in the objective space, and different colors indicate solutions after selection. The curves in **b** represent the statistical result values corresponding to the dimensions. **c** Shows the absolute difference between the two statistical models

Infill criterion

Once the d -dimensional decision vector set S , which has a more significant effect on convergence, is selected by adaptive dropout, the proposed ADSAPSO is expected to optimize the d decision variables and select promising candidate individuals to be re-evaluated.

In our proposed infill criterion, an RBF-based search is adapted to optimize the selected d -dimensional variables given in Algorithm 2. First, m d -dimensional RBF models are trained using the individuals in arc_d , which will be used to replace the original expensive functions for evaluating offspring. Notably, RBF is used due to its insensitivity to the increment in the dimension of the function to be approximated [32,33]. Compared with the Kriging model [34], RBF is more suitable for problems with high-dimensional decision variables, since the computation time for training Kriging models will become unbearable when the number of training samples is large. Then, the particle swarm opti-

Fig. 4 An example of the replacement

Algorithm 2 RBF-based Search

Input: NI (population size of PSO), λ_{max} (maximum number of iterations), arc_d (data to construct RBF), k (number of individuals to be re-evaluated), FES (function evaluations), $MaxFES$ (maximum function evaluations)

Output: X_d (k d -dimension vectors)

- 1: $RBF \leftarrow$ Train an RBF model for each objective function by using individuals in arc_d
- 2: $[Pbest, Gbest] \leftarrow$ Initialization(NI)
- 3: **for** $\lambda = 1 : \lambda_{max}$ **do**
- 4: $Off \leftarrow$ Generate offspring
- 5: **if** $FES \geq 0.75 \times MaxFES$ **then**
- 6: $Off \leftarrow$ Polynomial Mutation(Off)
- 7: **end if**
- 8: $Off \leftarrow$ Evaluate Off by RBF
- 9: $[Pbest, Gbest] \leftarrow$ Update($Off, Pbest, Gbest$)
- 10: **end for**

mizer (PSO) [35] is adopted for further optimization, due to its promising capability in solving high-dimensional optimization problems as suggested in Ref. [34]. To enhance population diversity at the late stage of the evolution, we add a polynomial mutation operation (Line 6 in Algorithm 2). In the late stage of the evolution, the population may trap in local optima easily, and thus $0.75 \times MaxFES$ is empirically adopted as the threshold for adopting polynomial mutation. After the optimization, k d -dimensional better-converged decision vectors are selected from the final decision vector set. Compared with the surrogate models built with D -dimensional training samples, although not all decision variables are optimized at every iteration, the surrogate model built with d -dimensional training samples can better help optimize the d -dimensional variables and accelerate convergence rate. The RBF-based search in the proposed ADSAPSO aims to conduct the local search in a low-dimensional space to quickly obtain better-converged solutions, which is naturally suitable for high-dimensional EMOPs.

Since the optimized k decision vectors of dimension d cannot be evaluated by expensive functions directly, we introduce a Replacement method that extends d -dimensional decision vectors to D -dimensional ones. To be more specific, k solutions of dimension D , named X_D , are first selected from the non-dominated solution set of arc by the environmental selection. Note that the environmental selection adopted here is the same as that in Algorithm 1. Next, the corresponding dimensions of X_D will be replaced by X_d to form k new individuals X_{new} . Specifically, solutions in X_d are well converged, and solutions in X_D are with good diversity. To some extent, the newly generated solutions X_{new} can be considered to inherit the convergence and diversity properties of X_d and X_D simultaneously. Finally, we re-evaluate X_{new} for updating the archives. An illustrative example of the Replacement is shown in Fig. 4, where each circle denotes a decision variable, and the dashed circles will be replaced for evaluating full-length decision vectors.

Experimental study

To empirically investigate the performance of the proposed ADSAPSO, we first discuss the impact of different parameter β value on the algorithm. Then, three state-of-the-art SAEAs, namely MOEA/D-EGO [19], K-RVEA [21], and RM-MEDA [36] are compared with our proposed ADSAPSO on test problems. Notably, we compare our proposed algorithm with RM-MEDA since both algorithms have adopted Statistical models during the optimization. In the rest of this section, we first present a brief introduction to the experimental settings of all the compared algorithms. Then the test problems and performance indicators are described. Afterwards, each algorithm is run 20 times on each test problem independently. The Wilcoxon rank sum test is used to compare the results obtained by the proposed ADSAPSO and the compared algorithms at a significance level of 0.05. Symbols “+”, “−”, and “ $\approx$ ” mean that the compared algorithm is significantly better

than, significantly worse than, and approximately equal to the proposed ADSAPSO, respectively.

Experimental settings

For fair comparisons, all the compared algorithms are implemented in PlatEMO [37] using MATLAB, and they are run on a computer with an Intel Core i7 processor and 32 GB of RAM.

- (1) Parameters settings: In our proposed ADSAPSO, the number of samples used to build the surrogate model is set to $N_\alpha = 200$, and the number of individuals to be re-evaluated is set to $k = 5$. The number of selecting for well-performing solutions and poorly-performing solutions in Solutions Selection is set to $N_s = 50$. In K-RVEA, the number of individuals to be re-evaluated and used for updating surrogate models is also set to 5. For the other compared algorithms, the recommended parameter settings in the literature are used.
- (2) Reproduction operators: In this paper, the maximum number of generations in the RBF-based search is set to $\lambda_{\max} = 100$. In PSO, the population size of PSO is set to $Nl = 100$, and the inertia weight of PSO is set to $W = 0.5$.
- (3) Termination Condition: For all the test problems except RM-MEDA, the size of the initial data is set to 500. The maximum function evaluations using the expensive function (denoted as MaxFEs) is set to 1000.

Test problems and performance indicator

In this paper, we use ten IMF problems selected from [38] and seven DTLZ problems. Among the IMF test suite and DTLZ test suite, the number of objectives is three in IMF4, IMF8 and all in DTLZ, and it is two in other problems.

The Inverted Generational Distance (IGD) indicator [39] is a commonly used performance indicator for multiobjective optimization, which can assess the quality of the obtained solution set from both the convergence and distribution uniformity. To better explain IGD, we suppose that P is a set of reference points that are evenly distributed on the Pareto-optimal Front (PF) and Q is the set of obtained non-dominated solutions. The mathematical definition of IGD is

$$IGD(P, Q) = \frac{\sum_{p \in P} \text{dis}(p, Q)}{|P|}, \quad (6)$$

where $\text{dis}(p, Q)$ is the minimum Euclidean distance between p and points in Q , and $|P|$ is the number of reference points in P . A smaller IGD value indicates better performance of the algorithm.

Impact of parameter N_s

In the proposed Adaptive Dropout Mechanism, parameter N_s is the number of selected well-performing solutions and poorly-performing solutions from the elite archive arc. A sensitivity analysis is conducted to investigate the impact of parameter N_s on the performance of ADSAPSO. Here, ADSAPSO with N_s being to 100 and 67 are compared with ADSAPSO with $N_s = 50$. Different N_s values are derived from the empirical choices where N_α is set to 200. $N_s = 100$ means that half of arc is selected, $N_s = 67$ indicates that the elite archive arc is divided into three equal parts, and $N_s = 50$ represents a quarter of the elite archive arc is obtained.

The statistical results of the IGD values achieved by ADSAPSO with different N_s values on the IMF test suite are summarized in Table 1. It can be observed that ADSAPSO with different N_s values performs similarly, and ADSAPSO with $N_s = 50$ has achieved the best results on these test problems. Thus, $N_s = 50$ is adopted in the rest experiments.

Impact of parameter β

Additional comparative experiments are run to investigate the impact of parameter β (indicating the probability of selecting the number of dimensions in the adaptive dropout mechanism). Keeping the same parameters settings as introduced in Sect. 4.1, the ADSAPSO with different β values from $\{0.1, 0.3, 0.5, 0.7, 1.0\}$ are empirically compared.

The statistical results of the IGD values achieved by ADSAPSO with different β values on the IMF test suite are summarized in Table 2.

It can be observed that ADSAPSO with $\beta = 0.5$ has achieved the best results on these test problems, and ADSAPSO with $\beta = 0.1$ has achieved the worst results, which indicates that different values of β have a significant influence on the result. In subsequent experiments, we will adopt $\beta = 0.5$ in our proposed ADSAPSO.

Simultaneously, $\beta = 1.0$ indicates that the adaptive dropout mechanism is not used in ADSAPSO, and full-length decision vectors are used during optimization. It can be observed that ADSAPSO with $\beta = 0.5$ shows an overall better performance than ADSAPSO without adaptive dropout mechanism. However, the ADSAPSO without adaptive dropout mechanism is better on IMF6 and IMF8. Comparing with ADSAPSO with $\beta = 1.0$, we can verify the effectiveness of our proposed adaptive dropout mechanism.

General performance

In this subsection, ADSAPSO is compared with two representative SAEAs, i.e., MOEA/D-EGO [19], K-RVEA [21], and RM-MEDA [36]. K-RVEA is an SAEA that relies on a set of adaptive reference vectors for expensive many-objective

Table 1 Statistics of IGD results obtained by ADSAPSO with different N_s on 30 IMF test instances

Problem	M	D	$N_s = 100$	$N_s = 67$	$N_s = 50$
IMF1	2	50	2.91e+0 (7.60e−1) −	2.89e+0 (7.97e−1) ≈	2.18e + 0 (4.47e − 1)
		100	3.89e+0 (6.12e−1) −	4.55e+0 (1.31e+0) −	3.13e+0 (6.25e − 1)
		200	6.20e+0 (8.69e−1) ≈	5.82e+0 (9.46e−1) ≈	5.55e + 0 (4.97e − 1)
IMF2	2	50	4.60e+0 (1.40e+0) ≈	5.17e+0 (9.68e−1) −	3.72e + 0 (8.60e − 1)
		100	6.72e+0 (7.86e−1) ≈	6.77e+0 (1.37e+0) ≈	5.93e + 0 (9.50e − 1)
		200	8.76e + 0 (4.94e − 1) ≈	8.84e+0 (1.20e+0) ≈	9.04e+0 (5.74e−1)
IMF3	2	50	4.63e+0 (8.21e−1) ≈	4.67e+0 (6.58e−1) ≈	4.15e + 0 (5.20e − 1)
		100	7.27e+0 (8.59e−1) −	7.30e+0 (1.06e+0) −	6.03e + 0 (2.74e − 1)
		200	9.00e+0 (9.00e−1) −	9.08e+0 (6.19e−1) −	8.19e + 0 (4.91e − 1)
IMF4	3	50	2.49e+1 (6.48e+0) ≈	2.34e+1 (3.86e+0) ≈	2.09e + 1 (4.72e + 0)
		100	6.27e+1 (9.97e+0) ≈	6.76e+1 (1.52e+1) ≈	5.73e + 1 (6.23e + 0)
		200	1.79e+2 (1.84e+1) ≈	1.92e+2 (3.69e+1) ≈	1.73e + 2 (1.30e + 1)
IMF5	2	50	2.41e−1 (7.51e−3) ≈	2.43e−1 (8.57e−3) ≈	2.38e − 1 (1.55e − 2)
		100	2.81e−1 (9.40e−3) ≈	2.73e − 1 (1.01e − 2) ≈	2.77e−1 (8.83e−3)
		200	2.96e−1 (1.08e−2) ≈	3.00e−1 (1.05e−2) ≈	2.92e − 1 (1.46e − 2)
IMF6	2	50	3.90e − 1 (2.31e − 2) ≈	4.19e−1 (2.59e−2) ≈	3.97e−1 (3.71e−2)
		100	4.46e−1 (4.34e−2) ≈	4.54e−1 (2.59e−2) ≈	4.39e − 1 (3.51e − 2)
		200	4.76e − 1 (3.85e − 2) ≈	5.19e−1 (3.83e−2) ≈	4.92e−1 (4.05e−2)
IMF7	2	50	3.68e−1 (2.70e−2) −	3.63e−1 (1.51e−2) −	3.46e − 1 (3.09e − 2)
		100	3.89e−1 (1.84e−2) −	3.94e−1 (2.49e−2) −	3.63e − 1 (3.02e − 2)
		200	3.87e − 1 (3.56e − 2) ≈	3.88e−1 (2.08e−2) ≈	3.94e−1 (3.84e−2)
IMF8	3	50	1.52e+0 (1.89e−1) −	1.45e+0 (1.63e−1) −	1.17e + 0 (3.41e − 1)
		100	2.65e+0 (4.15e−1) ≈	2.71e+0 (3.04e−1) ≈	2.41e + 0 (5.49e − 1)
		200	6.03e+0 (1.11e+0) ≈	5.93e + 0 (6.86e − 1) ≈	6.90e+0 (1.09e+0)
IMF9	2	50	3.75e−1 (3.08e−2) ≈	3.72e−1 (3.87e−2) ≈	3.68e − 1 (4.91e − 2)
		100	5.09e−1 (2.12e−2) ≈	4.98e − 1 (2.25e − 2) ≈	5.06e−1 (3.50e−2)
		200	5.70e−1 (6.50e−3) ≈	5.70e − 1 (4.83e − 3) ≈	5.71e−1 (8.05e−3)
IMF10	2	50	3.08e+2 (1.03e+2) ≈	2.40e + 2 (8.81e + 1) ≈	2.83e+2 (7.11e+1)
		100	7.28e+2 (1.20e+2) ≈	6.94e + 2 (9.91e + 1) ≈	7.11e+2 (1.37e+2)
		200	1.65e+3 (2.26e+2) ≈	1.50e + 3 (2.06e + 2) ≈	1.52e+3 (2.27e+2)
“+ / − / ≈”			0/7/23	0/7/23	

The best result in each row is highlighted

optimization, and the MOEA/D-EGO is an MOEA/D-based SAEA with the Gaussian stochastic process model. RM-MEDA is a regularity model-based multiobjective estimation of distribution algorithm, which is also based on statistical learning methods.

The experimental results of the IGD value provided by MOEA/D-EGO, K-RVEA, RM-MEDA, and ADSAPSO on

the IMF test suite and DTLZ test suite are recorded in Tables 3 and 4, respectively. Notably, MOEA/D-EGO fails to solve DTLZ4 with 200 decision variables due to the failure of Kriging models in handling high-dimensional data. It can be seen that ADSAPSO obtains better IGD values in most of the three test suite compared with MOEA/D-EGO, K-RVEA, and RM-MEDA.

Table 2 Statistics of IGD results obtained by ADSAPSO with different β on 30 IMF test instances

Problem	M	D	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.7$	$\beta = 1$	$\beta = 0.5$
IMF1	2	50	2.64e+0 (7.35e-1) $\approx$	1.80e + 0 (5.04e - 1) $\approx$	2.45e+0 (7.43e-1) $\approx$	3.38e+0 (8.02e-1) -	2.18e+0 (4.47e-1)
		100	3.44e+0 (6.84e-1) $\approx$	3.30e+0 (6.23e-1) $\approx$	4.75e+0 (5.91e-1) -	6.19e+0 (1.18e+0) -	3.13e + 0 (6.25e - 1)
		200	6.51e+0 (1.11e+0) $\approx$	5.85e+0 (8.85e-1) $\approx$	5.59e+0 (6.11e-1) $\approx$	7.58e+0 (1.83e+0) -	5.55e + 0 (4.97e - 1)
IMF2	2	50	4.33e+0 (1.10e+0) $\approx$	3.87e+0 (5.02e-1) $\approx$	3.79e+0 (8.79e-1) $\approx$	4.89e+0 (1.12e+0) -	3.72e + 0 (8.60e - 1)
		100	6.64e+0 (1.26e+0) $\approx$	6.11e+0 (7.93e-1) $\approx$	6.64e+0 (1.66e+0) $\approx$	6.71e+0 (1.32e+0) $\approx$	5.93e + 0 (9.50e - 1)
		200	8.75e+0 (1.08e+0) $\approx$	8.41e+0 (9.70e-1) $\approx$	8.28e + 0 (9.06e - 1) $\approx$	9.33e+0 (1.75e+0) $\approx$	9.04e+0 (5.74e-1)
IMF3	2	50	4.66e+0 (8.16e-1) $\approx$	4.65e+0 (8.79e-1) $\approx$	4.34e+0 (6.88e-1) $\approx$	4.55e+0 (5.62e-1) $\approx$	4.15e + 0 (5.20e - 1)
		100	7.20e+0 (1.46e+0) $\approx$	6.69e+0 (1.07e+0) $\approx$	5.92e + 0 (4.04e - 1) $\approx$	6.63e+0 (1.02e+0) $\approx$	6.03e+0 (2.74e-1)
		200	9.49e+0 (7.52e-1) -	8.51e+0 (1.13e+0) $\approx$	8.70e+0 (1.52e+0) $\approx$	9.68e+0 (1.00e+0) -	8.19e + 0 (4.91e - 1)
IMF4	3	50	2.74e+1 (5.05e+0) -	2.21e+1 (3.37e+0) $\approx$	2.27e+1 (4.40e+0) $\approx$	2.22e+1 (4.79e+0) $\approx$	2.09e + 1 (4.72e + 0)
		100	8.15e+1 (1.27e+1) -	6.71e+1 (1.15e+1) $\approx$	6.04e+1 (1.11e+1) $\approx$	7.35e+1 (1.27e+1) -	5.73e + 1 (6.23e + 0)
		200	2.26e+2 (4.46e+1) -	1.76e+2 (4.00e+1) $\approx$	1.82e+2 (1.79e+1) $\approx$	2.43e+2 (4.52e+1) -	1.73e + 2 (1.30e + 1)
IMF5	2	50	2.46e-1 (3.51e-3) $\approx$	2.51e-1 (8.71e-3) $\approx$	2.47e-1 (1.08e-2) $\approx$	2.42e-1 (1.42e-2) $\approx$	2.38e - 1 (1.55e - 2)
		100	2.78e-1 (6.43e-3) $\approx$	2.78e-1 (7.73e-3) $\approx$	2.79e-1 (1.71e-2) $\approx$	2.72e - 1 (1.41e - 2) $\approx$	2.77e-1 (8.83e-3)
		200	2.92e-1 (7.03e-3) $\approx$	3.02e-1 (4.25e-3) -	3.01e-1 (9.25e-3) $\approx$	2.79e - 1 (1.60e - 2) $\approx$	2.92e-1 (1.46e-2)
IMF6	2	50	3.84e-1 (1.16e-2) $\approx$	4.14e-1 (2.12e-2) $\approx$	3.98e-1 (2.52e-2) $\approx$	3.73e - 1 (4.94e - 2) $\approx$	3.97e-1 (3.71e-2)
		100	4.42e-1 (2.55e-2) $\approx$	4.70e-1 (1.17e-2) -	4.66e-1 (4.01e-2) $\approx$	4.46e-1 (3.21e-2) $\approx$	4.39e - 1 (3.51e - 2)
		200	4.99e-1 (2.14e-2) $\approx$	5.08e-1 (1.89e-2) $\approx$	4.78e-1 (4.00e-2) $\approx$	4.05e - 1 (2.37e - 2) +	4.92e-1 (4.05e-2)
IMF7	2	50	3.64e-1 (1.50e-2) -	3.61e-1 (1.83e-2) $\approx$	3.59e-1 (3.41e-2) $\approx$	3.58e-1 (2.43e-2) $\approx$	3.46e - 1 (3.09e - 2)
		100	4.06e-1 (2.43e-2) -	4.03e-1 (2.98e-2) -	3.87e-1 (3.89e-2) $\approx$	3.66e-1 (1.99e-2) $\approx$	3.63e - 1 (3.02e - 2)
		200	4.33e-1 (1.50e-2) -	3.74e - 1 (2.21e - 2) $\approx$	3.94e-1 (3.71e-2) $\approx$	4.04e-1 (4.41e-2) $\approx$	3.94e-1 (3.84e-2)
IMF8	3	50	1.78e+0 (1.20e-1) -	1.59e+0 (1.30e-1) -	1.23e+0 (8.57e-2) $\approx$	9.48e - 1 (3.81e - 2) $\approx$	1.17e+0 (3.41e-1)
		100	3.44e+0 (4.61e-1) -	3.23e+0 (6.82e-1) -	2.22e+0 (1.97e-1) $\approx$	1.65e + 0 (3.24e - 1) +	2.41e+0 (5.49e-1)
		200	8.11e+0 (5.63e-1) -	7.30e+0 (6.35e-1) $\approx$	5.03e+0 (3.19e-1) +	3.61e + 0 (1.24e + 0) +	6.90e+0 (1.09e+0)

Table 2 continued

Problem	M	D	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.7$	$\beta = 1$	$\beta = 0.5$
IMF9	2	50	4.33e-1 (5.15e-2) –	4.10e-1 (3.99e-2) $\approx$	4.43e-1 (3.85e-2) –	4.69e-1 (5.63e-2) –	3.68e – 1 (4.91e – 2)
		100	5.26e-1 (2.05e-2) $\approx$	5.18e-1 (1.96e-2) $\approx$	5.11e-1 (1.97e-2) $\approx$	5.44e-1 (8.67e-3) –	5.06e – 1 (3.50e – 2)
		200	5.77e-1 (5.15e-3) $\approx$	5.77e-1 (1.95e-3) –	5.66e – 1 (5.43e – 3) $\approx$	5.81e-1 (4.30e-3) –	5.71e-1 (8.05e-3)
IMF10	2	50	3.00e+2 (2.25e+1) $\approx$	3.31e+2 (4.08e+1) $\approx$	2.39e + 2 (6.73e + 1) $\approx$	3.08e+2 (4.78e+1) $\approx$	2.83e+2 (7.11e+1)
		100	7.30e+2 (1.05e+2) $\approx$	7.81e+2 (8.27e+1) $\approx$	6.79e + 2 (1.65e + 2) $\approx$	7.04e+2 (1.07e+2) $\approx$	7.11e+2 (1.37e+2)
		200	1.69e+3 (1.30e+2) –	1.82e+3 (4.29e+1) $\approx$	1.53e+3 (2.11e+2) $\approx$	1.80e+3 (1.37e+2) –	1.52e + 3 (2.27e + 2)
“+/-/ $\approx$ ”			0/12/18	0/6/24	1/2/27	3/11/16	

The best result in each row is highlighted

Table 3 Statistics of IGD results obtained by MOEA/D-EGO, K-RVEA, RM-MEDA and ADSAPSO on 30 IMF test instances

Problem	<i>M</i>	<i>D</i>	MOEA/D-EGO	K-RVEA	RM-MEDA	ADSAPSO
IMF1	2	50	7.60e+0 (1.43e+0) –	2.94e+0 (3.75e+0) ≈	4.49e+0 (6.45e−1) –	2.18e + 0 (4.47e − 1)
[3pt]		100	1.06e+1 (1.86e+0) –	1.14e+1 (7.81e−1) –	6.30e+0 (7.00e−1) –	3.13e + 0 (6.25e − 1)
[3pt]		200	1.31e+1 (3.94e−1) –	1.32e+1 (6.52e−1) –	8.64e+0 (1.09e+0) –	5.55e + 0 (4.97e − 1)
[3pt] IMF2	2	50	1.05e+1 (1.36e+0) –	3.17e + 0 (4.54e + 0) ≈	6.31e+0 (1.01e+0) –	3.72e+0 (8.60e−1)
[3pt]		100	1.17e+1 (2.02e+0) –	1.35e+1 (4.57e+0) –	8.42e+0 (1.18e+0) –	5.93e + 0 (9.50e − 1)
[3pt]		200	1.63e+1 (4.46e−1) –	1.72e+1 (5.17e−1) –	1.16e+1 (8.88e−1) –	9.04e + 0 (5.74e − 1)
[3pt] IMF3	2	50	8.96e+0 (1.83e+0) –	7.03e+0 (7.15e+0) ≈	7.65e+0 (9.14e−1) –	4.15e + 0 (5.20e − 1)
[3pt]		100	1.26e+1 (1.71e+0) –	1.46e+1 (1.32e+0) –	1.09e+1 (1.14e+0) –	6.03e + 0 (2.74e − 1)
[3pt]		200	1.66e+1 (8.90e−1) –	1.65e+1 (7.61e−1) –	1.35e+1 (1.08e+0) –	8.19e + 0 (4.91e − 1)
[3pt] IMF4	3	50	5.47e+1 (7.88e+0) –	2.01e + 1 (2.26e + 1) ≈	4.67e+1 (7.51e+0) –	2.09e+1 (4.72e+0)
[3pt]		100	1.51e+2 (1.95e+1) –	8.69e+1 (6.01e+1) ≈	1.33e+2 (8.20e+0) –	5.73e + 1 (6.23e + 0)
[3pt]		200	3.60e+2 (7.35e+0) –	3.57e+2 (2.71e+1) –	3.15e+2 (3.04e+1) –	1.73e + 2 (1.30e + 1)
[3pt] IMF5	2	50	2.52e−1 (7.88e−3) –	1.70e − 1 (2.00e − 2)+	2.28e−1 (8.76e−3) ≈	2.38e−1 (1.55e−2)
[3pt]		100	2.87e−1 (8.49e−3) –	2.88e−1 (1.37e−2) –	2.63e − 1 (1.13e − 2)+	2.77e−1 (8.83e−3)
[3pt]		200	3.04e−1 (9.99e−3) ≈	3.03e−1 (7.40e−3) ≈	2.87e − 1 (5.60e − 3) ≈	2.92e−1 (1.46e−2)
[3pt] IMF6	2	50	3.93e−1 (3.81e−2) ≈	3.37e − 1 (4.30e − 2)+	3.59e−1 (2.13e−2) +	3.97e−1 (3.71e−2)
[3pt]		100	4.95e−1 (2.11e−2) –	4.98e−1 (2.91e−2) –	4.30e − 1 (2.19e − 2) ≈	4.39e−1 (3.51e−2)
[3pt]		200	5.42e−1 (1.49e−2) –	5.39e−1 (8.24e−3) –	4.88e − 1 (4.27e − 2) ≈	4.92e−1 (4.05e−2)
[3pt] IMF7	2	50	3.62e−1 (2.24e−2) ≈	3.46e−1 (3.10e−2) ≈	3.57e−1 (1.09e−2) –	3.46e − 1 (3.09e − 2)
[3pt]		100	4.29e−1 (1.58e−2) –	4.42e−1 (1.47e−2) –	3.87e−1 (1.18e−2) –	3.63e − 1 (3.02e − 2)
[3pt]		200	4.49e−1 (2.72e−2) –	4.78e−1 (6.56e−3) –	4.23e−1 (1.78e−2) ≈	3.94e − 1 (3.84e − 2)
[3pt] IMF8	3	50	1.65e+0 (2.41e−1) –	1.43e+0 (1.03e−1) –	1.38e+0 (1.66e−1) ≈	1.17e + 0 (3.41e − 1)
[3pt]		100	4.29e+0 (2.11e−1) –	3.78e+0 (4.98e−1) –	2.86e+0 (6.19e−1) ≈	2.41e + 0 (5.49e − 1)
[3pt]		200	9.41e+0 (3.77e−1) –	9.16e+0 (5.06e−1) –	5.90e + 0 (6.44e − 1)+	6.90e+0 (1.09e+0)
IMF9	2	50	4.99e−1 (3.04e−2) –	4.48e−1 (2.48e−2) –	5.13e−1 (8.59e−3) –	3.68e − 1 (4.91e − 2)
		100	5.55e−1 (1.66e−3) –	5.48e−1 (2.26e−2) –	5.51e−1 (4.30e−3) –	5.06e − 1 (3.50e − 2)
		200	5.93e−1 (1.92e−3) –	5.95e−1 (1.22e−3) –	5.87e−1 (2.40e−3) –	5.71e − 1 (8.05e − 3)
IMF10	2	50	3.71e+2 (2.41e+1) –	4.04e+2 (5.58e+1) –	3.51e+2 (2.97e+1) ≈	2.83e + 2 (7.11e + 1)
		100	9.49e+2 (5.90e+1) –	9.69e+2 (3.70e+1) –	8.50e+2 (2.57e+1) –	7.11e + 2 (1.37e + 2)
		200	2.07e+3 (1.52e+2) –	2.10e+3 (8.08e+1) –	1.80e+3 (1.03e+2) –	1.52e + 3 (2.27e + 2)
“+/-/≈”			0/27/3	2/21/7	3/19/8	

The best result in each row is highlighted

Table 4 Statistics of IGD results obtained by MOEA/D-EGO, K-RVEA, RM-MEDA, and ADSAPSO on 21 DTLZ test instances

Problem	M	D	MOEA/D-EGO	K-RVEA	RM-MEDA	ADSAPSO
DTLZ1	3	50	1.00e+3 (2.09e+2) –	1.08e+3 (9.91e+1) –	8.16e+2 (6.96e+1) –	6.64e + 2 (1.44e + 2)
		100	2.65e+3 (5.13e+1) –	2.55e+3 (1.84e+2) –	1.89e + 3 (1.32e + 2) ≈	1.92e+3 (2.39e+2)
		200	5.66e+3 (1.19e+2) –	5.64e+3 (2.47e+2) –	4.01e + 3 (2.81e + 2)+	4.65e+3 (3.83e+2)
DTLZ2	3	50	2.01e+0 (2.88e−1) –	1.26e+0 (1.24e−1) –	1.51e+0 (1.13e−1) –	7.48e − 1 (1.40e − 1)
		100	6.17e+0 (1.84e−1) –	5.30e+0 (5.05e−1) –	3.47e+0 (2.85e−1) –	2.21e + 0 (6.87e − 1)
		200	1.36e+1 (2.84e−1) –	1.36e+1 (3.04e−1) –	7.33e+0 (1.66e−1) –	6.04e + 0 (1.45e + 0)
DTLZ3	3	50	2.28e+3 (8.32e+2) –	2.98e+3 (2.57e+2) –	2.36e+3 (1.97e+2) –	1.39e + 3 (2.36e + 2)
		100	7.55e+3 (6.03e+2) –	8.15e+3 (4.14e+2) –	5.58e+3 (2.97e+2) –	4.12e + 3 (5.98e + 2)
		200	1.60e+4 (1.17e+3) –	1.83e+4 (2.85e+2) –	1.18e+4 (6.33e+2) –	8.34e + 3 (1.18e + 3)
DTLZ4	3	50	2.48e+0 (2.29e−1) –	3.07e+0 (1.60e−1) –	1.82e+0 (8.55e−2) –	6.78e − 1 (1.63e − 1)
		100	4.78e+0 (7.07e−1) –	6.49e+0 (2.77e−1) –	3.85e+0 (3.23e−1) –	2.22e + 0 (3.06e − 1)
		200	/	1.39e+1 (2.66e−1) –	7.85e+0 (6.70e−1) –	5.93e + 0 (1.42e + 0)
DTLZ5	3	50	1.81e+0 (3.47e−1) –	1.36e+0 (1.42e−1) –	1.63e+0 (1.23e−1) –	8.46e − 1 (1.30e − 1)
		100	5.77e+0 (7.63e−1) –	5.38e+0 (7.14e−1) –	3.43e+0 (2.02e−1) –	2.19e + 0 (4.05e − 1)
		200	1.31e+1 (8.08e−1) –	1.34e+1 (3.49e−1) –	7.53e+0 (6.88e−1) –	5.08e + 0 (1.33e + 0)
DTLZ6	3	50	2.48e+1 (4.44e+0) –	3.12e+1 (1.29e+0) –	2.88e+1 (1.70e+0) –	2.11e + 1 (2.70e + 0)
		100	6.50e+1 (7.23e+0) –	8.47e+1 (1.77e+0) –	6.35e+1 (2.72e+0) –	4.86e + 1 (6.54e + 0)
		200	1.62e+2 (5.24e+0) –	1.76e+2 (3.97e−1) –	1.28e+2 (4.47e+0) –	1.11e + 2 (1.03e + 1)
DTLZ7	3	50	6.85e+0 (1.40e+0) –	4.55e+0 (4.52e+0) ≈	8.18e+0 (3.93e-1) –	3.74e + 0 (1.20e + 0)
		100	9.44e+0 (1.16e+0) –	9.26e+0 (2.43e+0) –	9.34e+0 (4.55e−1) –	5.36e + 0 (1.32e + 0)
		200	1.06e+1 (2.69e-1) –	1.04e+1 (6.40e-1) –	1.02e+1 (2.34e−1) –	7.08e + 0 (4.91e − 1)
“+/-/≈”			0/20/0	0/20/1	1/19/1	

The best result in each row is highlighted. “/”: Fail to solve the problem

The convergence profiles of the three compared algorithms on DTLZ problems with 200 decision variables are given in Fig. 5.

It can be observed that our proposed ADSAPSO converges faster than the other three compared algorithms on most problems. The results have demonstrated the superiority of our proposed ADSAPSO over the three compared algorithms on EMOPs in terms of convergence speed.

Figure 6 presents the archive *Arc* achieved by the ADSAPSO, MOEA/D-EGO and K-RVEA on bi-objective DTLZ1 with 50 decision variables at 500 evaluations (Initial archive), 750 evaluations (Interim archive), and 1000 evaluations (Final archive) to visualize the experimental results more intuitively.

It can be observed that the archive obtained from ADSAPSO converges in several directions at a faster convergence rate in the early stage, and the convergence slows down in the later stage, but the population diversity increases. MOEA/D-EGO and K-RVEA hardly converge in the whole process. Overall,

ADSAPSO has achieved the best results on these problems, where the obtained final archive *Arc* are best converged.

Conclusion

In this paper, we have proposed an SAEA with adaptive dropout mechanism, called ADSAPSO, for solving high-dimensional EMOPs with up to 200 decision variables.

Generally, the adaptive dropout mechanism is used for selecting *d*-dimension decision variables from *D*-dimensional original decision variables. It can help build *d*-dimensional relatively high-precision RBF models, and the algorithm only needs to optimize the *d*-dimensional variables at each iteration. The infill criterion is adopted to select candidate individuals for re-evaluation, where the corresponding dimensions in *k* *D*-dimensional solutions are replaced by the *k* *d*-dimensional optimized individuals to form *k* new *D*-dimensional individuals.

Fig. 5 Convergence profiles of the three compared algorithms on DTLZ1–DTLZ7 with 200 decision variables

Some systematic comparisons have been conducted on a set of EMOPs with up to 200 decision variables. First, we discuss the impact of different parameter β value on the algorithm and find $\beta = 0.5$ is the best. The proposed ADSAPSO is compared with three representative algorithms, i.e., MOEA/D-EGO, K-RVEA, and RM-MEDA, on test problems IMF and DTLZ. The experimental results indicate the superiority of ADSAPSO in solving EMOPs with high-dimension decision variables.

This paper indicates that ADSAPSO is promising in solving high-dimension EMOPs. Furthermore, it is promising to extend ADSAPSO to solving expensive large-scale multiobjective optimization problems [40] and expensive many-objective optimization problems. It is also desirable to apply it to real-world applications.

Fig. 6 Arc obtained by the compared algorithms on bi-objective DTLZ1 with 50 decision variables at different stages of the evolution

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 61903178, 61906081, and U20A20306), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), the Shenzhen Peacock Plan (Grant No. KQTD2016112514355531), and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. He C, Cheng R, Zhang C, Tian Y, Chen Q, Yao X (2020) Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers. *IEEE Trans Evol Comput* 24(5):868–881
2. Pan L, He C, Tian Y, Su Y, Zhang X (2017) A region division based diversity maintaining approach for many-objective optimization. *Integr Comput Aided Eng* 24(3):279–296
3. Lu Z, Whalen I, Dhebar Y et al (2020) Multi-objective evolutionary design of deep convolutional neural networks for image classification. *IEEE Trans Evol Comput* 25(2):277–291
4. Sun G, Pang T, Fang J, Li G, Li Q (2017) Parameterization of criss-cross configurations for multiobjective crashworthiness optimization. *Int J Mech Sci* 124:145–157
5. Abido MA (2006) Multiobjective evolutionary algorithms for electric power dispatch problem. *IEEE Trans Evol Comput* 10(3):315–329
6. He C, Tian Y, Jin Y, Zhang X, Pan L (2017) A radial space division based many-objective optimization evolutionary algorithm. *Appl Soft Comput* 61:603–621

7. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
8. He C, Cheng R, Danial Y (2020) Adaptive offspring generation for evolutionary large-scale multiobjective optimization. *IEEE Trans Syst Man Cybern Syst*. <https://doi.org/10.1109/TSMC.2020.3003926>
9. He C, Cheng R, Jin Y, Yao X (2019) Surrogate-assisted expensive many-objective optimization by model fusion. In: 2019 IEEE congress on evolutionary computation (CEC), pp 1672–1679. IEEE
10. Akhtar T, Shoemaker CA (2016) Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *J Glob Optim* 64(1):17–32
11. Pan L, He C, Tian Y, Wang H, Zhang X, Jin Y (2019) A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput* 23(1):74–88
12. He C, Huang S, Cheng R, Tan KC, Jin Y (2020) Evolutionary multiobjective optimization driven by generative adversarial networks (GANs). *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2020.2985081>
13. Zhou Z, Ong Y, Nguyen M, Lim D (2005) A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In: 2005 IEEE congress on evolutionary computation, vol 3, pp 2832–2839
14. 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
15. Wang Y, Yin D, Yang S, Sun G (2019) Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. *IEEE Trans Cybern* 49(5):1642–1656
16. Regis RG (2014) Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. *IEEE Trans Evol Comput* 18(3):326–347
17. Jin Y (2011) Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput* 1(2):61–70
18. Knowles J (2006) Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans Evol Comput* 10(1):50–66
19. Zhang Q, Liu W, Tsang E, Virginas B (2010) Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans Evol Comput* 14(3):456–474
20. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
21. Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans Evol Comput* 22(1):129–142
22. Hussein R, Deb K (2016) A generative kriging surrogate model for constrained and unconstrained multi-objective optimization. *Proc Genet Evol Comput Conf* 2016:573–580
23. Zhao M, Zhang K, Chen G, Zhao X, Yao C, Sun H, Huang Z, Yao J (2020) A surrogate-assisted multi-objective evolutionary algorithm with dimension-reduction for production optimization. *J Pet Sci Eng* 192:107192
24. Li C, Gupta S, Rana S, Nguyen V, Venkatesh S, Shilton A (2017) High dimensional Bayesian optimization using dropout. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, pp 2096–2102
25. Chen G, Zhang K, Xue X, Zhang L, Yao J, Sun H, Fan L, Yang Y (2020) Surrogate-assisted evolutionary algorithm with dimensionality reduction method for water flooding production optimization. *J Pet Sci Eng* 185:106633
26. Li F, Cai X, Gao L, Shen W (2021) A surrogate-assisted multi-swarm optimization algorithm for high-dimensional computationally expensive problems. *IEEE Trans Cybern* 51(3):1390–1402
27. Guo D, Wang X, Gao K, Jin Y, Ding J, Cai T (2021) Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. In: *IEEE transactions on systems, man, and cybernetics: systems*, pp 1–14
28. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. *Comput Sci* 3(4):212–223
29. Peter M (2002) What is a statistical model? *Ann Stat* 30(5):1225–1310
30. Stein M (1987) Large sample properties of simulations using latin hypercube sampling. *Technometrics* 29(2):143–151
31. Ma M, Li H, Huang J (2018) A multi-objective evolutionary algorithm based on principal component analysis and grid division. In: 2018 14th international conference on computational intelligence and security (CIS), pp 201–204
32. Er M, Wu S, Lu J, Toh H (2002) Face recognition with radial basis function (RBF) neural networks. *IEEE Trans Neural Netw* 13(3):697–710
33. Kattan A, Galvan E (2012) Evolving radial basis function networks via GP for estimating fitness values using surrogate models. In: 2012 IEEE congress on evolutionary computation, pp 1–7
34. Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21(4):644–660
35. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95—international conference on neural networks*, vol 4, pp 1942–1948
36. Zhang Q, Zhou A, Jin Y (2008) Rm-meda: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput* 12(1):41–63
37. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: a matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87
38. Cheng R, Jin Y, Narukawa K, Sendhoff B (2015) A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling. *IEEE Trans Evol Comput* 19(6):838–856
39. Yen GG, He Z (2014) Performance metric ensemble for multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 18(1):131–144
40. He C, Li L, Tian Y, Zhang X, Cheng R, Jin Y, Yao X (2019) Accelerating large-scale multiobjective optimization via problem reformulation. *IEEE Trans Evol Comput* 23(6):949–961

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.