Complex & Intelligent Systems (2024) 10:171–191
https://doi.org/10.1007/s40747-023-01168-3

ORIGINAL ARTICLE

Dual-drive collaboration surrogate-assisted evolutionary algorithm by coupling feature reduction and reconstruction

Haibo Yu[2] · Yiyun Gong[1] · Li Kang[3] · Chaoli Sun[4] · Jianchao Zeng[1,2]

Received: 11 March 2023 / Accepted: 29 June 2023 / Published online: 17 July 2023

Abstract

Surrogate-assisted evolutionary algorithm (SAEA) prevails in the optimization of computationally expensive problems. However, existing SAEAs confront low efficiency in the resolution of high-dimensional problems characterized by multiple local optima and multivariate coupling. To this end, this paper offers a dual-drive collaboration surrogate-assisted evolutionary algorithm (DDCSAEA) by coupling feature reduction and reconstruction, which coordinates two unsupervised feature learning techniques, i.e., principal component analysis and autoencoder, in tandem. DDCSAEA creates a low-dimensional solution space by downscaling the target high-dimensional space via principal component analysis and collects promising candidates in the reduced space by collaborating a surrogate-assisted evolutionary sampling with differential mutation. An autoencoder is used to perform the feature reconstruction on the collected candidates for infill-sampling in the target high-dimensional space to sequentially refine the neighborhood landscapes of the optimal solution. Experimental results reveal that DDCSAEA has stronger convergence performance and optimization efficiency against eight state-of-the-art SAEAs on high-dimensional benchmark problems within 200 dimensions.

Keywords Expensive problems · Surrogate model · Principal component analysis · Autoencoder · Evolutionary algorithms · Feature reduction and reconstruction

Introduction

Population-driven evolutionary algorithms (EAs) have attracted considerable interest and applications in complex black-box system design and optimization over the past several decades. Their applications include but are not limited to the design optimization of spacecraft and chemical reactors [1, 2], neural network architectures [3, 4], morphological topology [5, 6], at so forth, because of their simple principles, ease of operation, robustness, and especially the weak dependence on the problem attributes. Nonetheless, to locate the optimal solution to the optimization problem, EAs usually necessitate triggering a large number of objective function evaluations. This makes them inefficient and computationally expensive in the resolution of problems involving time-consuming high-precision simulation and analysis, such as computational fluid dynamics, finite element analysis, and physical and chemical experiments, significantly hindering their practicality [7]. To break the performance bottleneck of EAs in dealing with computationally expensive optimization problems, SAEAs have gained wide attention. SAEA boosts the efficiency of EA in resolving computationally expensive optimization problems by constructing an inexpensive surrogate model to approximately replace the expensive target fitness function for fitness evaluation, which considerably boosts the optimization efficiency of EA [8–10]. The commonly used surrogate models mainly include Radial Basis Function (RBF) [11–13], Kriging or Gaussian Process (GP) [12–14], Support Vector Machine (SVM) [15, 16], and Polynomial Regression (PR) [17], etc.

✉ Haibo Yu
tyustyuhaibo@126.com

✉ Jianchao Zeng
20160002@nuc.edu.cn

[1] School of Electrical and Control Engineering, North University of China, Taiyuan 030051, China

[2] Institute of Big Data and Visual Computing, North University of China, Taiyuan 030051, China

[3] School of Environment and Safety Engineering, North University of China, Taiyuan 030051, China

[4] School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

Nowadays, the focuses of SAEAs concern the surrogate modeling for different types of optimization problems or underlying algorithms and the customization of effective model management strategies to optimally regulate the balance between the frequency of target fitness function invocations and solution accuracy on maximizing the algorithm performance. Model management is critical for the correct convergence of these methodologies. The primary components of model management cover the cooperative operation of the surrogate model with the EA's learning operators and the impact of infill samples on the model's correctness during the optimization process. To improve the training quality and prediction accuracy of the surrogate model, inspired by the Tri-Training semi-supervised learning technique, Wang et al. [18] proposed a new surrogate-assisted multi-objective optimization algorithm MOO-TTSA. The proposal used the Tri-Training in each iteration to filter the samples with higher confidence in fitness among the candidate solutions to enlarge the training sample set and optimize the modeling quality of the surrogate. Tong et al. [19] calculated the leave-one-out cross-validation error of the training sample set and constructed the uncertainty prediction model of candidate solutions based on the RBF model. The samples with the best-approximated fitness and the largest uncertainty in the design space were chosen for infill sampling, respectively. By leveraging the feature selection and feature extraction techniques in parallel, Guo et al. [20] enriched the training samples with three different feature attributes and constructed an ensemble model to approximate the target solution space landscape. The lower confidence bound and the expected improvement acquisition functions were also improved depending on the prediction variance of the three base models. In addition, considering the diversity and accuracy of surrogate ensemble modeling in the solution space, Yu et al. [21] calculated the prediction error sum of squares (PRESS) for RBF models paired with five different kernel functions by cross-validation techniques. Two RBF base models with optimal PRESS were selected to construct an ensemble model to estimate the fitness of candidate solutions. To improve the accuracy of the surrogate model and balance the exploration and exploitation of the algorithm, the best and the worst individuals in the iterative population were selected for real evaluation at each iteration, respectively.

However, as the problem scale and complexity increase, the issue of the curse of dimensionality, arising from high-dimensional complex feature space with multiple local optima and multivariate coupling, causes the exponential growth in the demand for the training sample size for surrogate modeling. As a result, the training cost and overfitting risk of the surrogate model increase, which greatly limits the effectiveness of SAEA. To improve the computational efficiency of SAEAs for high-dimensional computationally expensive optimization problems, spatial transformation and dimensional learning techniques have gained great attention. For reducing the training cost of GP models in high-dimensional decision space, Sammon mapping was employed in [12] to downscale the high-dimensional feature space to a low-dimensional one, and the GP models were then constructed in the reduced feature subspace to screen promising solutions in conjunction with the lower confidence bound criterion. Similarly, the high-dimensional feature space was also approximately simplified by the Sammon mapping in [22], and the iterative population was dynamically assigned with different differential mutation operators to generate offspring individuals based on the feedback information of the state of the optimal solution. Meanwhile, the global or local GP models were opted for and constructed in the feature subspace for different mutation strategies. In [23], an eigen coordinate system associated with the original coordinate system was generated through spatial transformation, and these two coordinate systems were then taken collaboratively with RBF-based multi-swam optimization to generate new candidate populations on a certain probability. When it comes to complex large-scale expensive optimization problems, the principal component analysis (PCA) was employed in [24] to help simplify the modeling complexity of the GP model by linearly mapping the training samples to a low-dimensional subspace, so that each objective function can be well approximated by the GP model to direct the optimization more accurately, and the solutions with the smallest angle-penalized distances and the largest uncertainties were chosen for subsequent refreshing of the GP models. Inspired by the divide-and-conquer philosophy, at each generation, the large-scale solution space was reduced to a series of low-dimensional subspaces using PCA and a random feature selection technique [25], and an adaptive search switch strategy was used to regulate the search of the subspaces at different optimization stages, allowing the iterative population and surrogate model to better accommodate the potential exploration and exploitation directions offered by the subspaces of the original and mapping spaces. A concept based on transfer optimization was adopted in [26], wherein a simplified problem space was constructed for the target problem feature space in line with the principal component analysis (PCA). The simplified problem space and the mapping relationship matrix between the two spaces are periodically reconstructed and updated according to the squared reconstruction error to ensure a positive transfer of the optimal information during the bi-spatial search. In [27] opted for a feature selection technique to condense the large-scale decision space and local surrogates were trained to approximate the landscape of the resulting low-dimensional feature subspace. Unlike the aforementioned approaches, the high-dimensional feature space was compressed and reconstructed via the encoding and decoding operators of an autoencoder in [28]. Two variable-size subpopulations were co-evolved and communicated in the original high-dimensional feature space and approximated low-dimensional feature subspace, significantly improving the solution efficiency of the SAEA on high-dimensional expensive optimization problems.

The usage of dimension reduction techniques to reduce the original complex high-dimensional feature space into a lower, more easily solvable feature subspace, makes it possible to control the complexity of the surrogate modeling on one hand. On the other hand, it can greatly increase the optimization efficiency of the SAEAs for high-dimensional solution space. Nevertheless, the loss of feature information associated with dimension reduction often results in a mismatch of the optimal structural properties between the feature subspace and the original feature space, which directly affects the accuracy and precision of SAEAs. In fact, in the framework of SAEAs combined with dimension reduction techniques, the mapping relationship model between the original feature space and the feature subspace is often built based on a small amount of historical evolutionary samples. The prediction accuracy of the mapping relationship model is strongly dependent on the quality and distribution of the historical evolutionary samples. Meanwhile, the candidate solutions derived from the feature subspace optimization are usually subject to inverse mapping to reconstruct features for generating solutions to be evaluated in the original feature space. The prediction quality of the mapping relationship model therefore directly determines the correctness of subsequent candidate solution screening and evaluation. Currently, existing SAEAs with dimension reduction techniques usually use the derived eigenvector matrix from the training samples for feature reconstruction for the newly generated candidates. Given the quality discrepancies and distribution characteristics of the training samples, directly adopting the derived eigenvector matrix for feature reconstruction of the newly added candidates can easily lead to feature drift, deteriorating the quality of the candidate samples in the original feature space and misleading the convergence direction of the iterative population. To address the above concerns, this paper leverages two unsupervised feature learning techniques, i.e., principal component analysis and autoencoder, to perform feature reduction and feature reconstruction of the high-dimensional solution space, and proposes a dual-drive collaboration SAEA, named DDCSAEA, for high-dimensional expensive optimization problems. The proposal's main contributions are as follows.

(1) A new feature reduction-driven surrogate-assisted subspace search strategy is proposed to simplify the surrogate modeling complexity and extract the principal component prior to the optimal solution during the iteration, based on the PCA and an RBF-assisted local search.

(2) A new feature reconstruction-driven infill-sampling strategy is designed for reconstructing and filtering the promising solutions of the feature subspace for real evaluation in the target problem space, by taking advantage of differential mutation and an autoencoder.

(3) A comprehensive analysis concerning the performance discrepancy of SAEAs under the single and sequential coupling modes of feature reduction and feature reconstruction is provided. The contrastive results show that the proposed method has better robustness and remarkable performance over five state-of-the-art algorithms on high-dimensional complex problems with multi-type fitness landscapes.

The remainder of the paper is structured as follows: "Related work" briefly introduces the principles of the RBF model, PCA, Autoencoder and the underlying local search engine. "Dual-drive collaboration surrogate-assisted evolutionary algorithm by coupling feature reduction and reconstruction" provides the motivation and a detailed description of the proposed method. "Empirical study" gives the experimental results and analyses. Finally, "Conclusion" concludes the paper and discusses some future works.

Related work

Social learning particle swarm optimizer

DDCSAEA employs the social learning particle swarm optimizer (SLPSO) to be the local search engine for the exploration of the optimal solution $\mathbf{Gbest}_{RBF}$ of the RBF in the feature subspace. SLPSO being a new PSO variant uses the randomly selected excellent exemplars and the mean position of the iterative swarm to replace the personal best **Pbest** and the swarm best **Gbest**, respectively, to guide the behavior learning of particles. Specifically, during the behavior learning of SLPSO, the iterative swarm members are first ranked in ascending order of fitness. The smaller the rank, the better the fitness. Then for each learner (particle), an exemplar is figured out from the swarm members with better ranks against the learner to guide its behavior learning. Here, the best particle in the current iterative swarm is directly retained in the new iterative swarm without undergoing behavior learning. The experimental results show that the SLPSO performs excellently in handling complex optimization problems.

Without loss of generality, for the minimization problem, SLPSO updates the velocities and positions of the particles with Eqs. (1) and (2), respectively.

$$\Delta x_{ij}^{(t+1)} = \gamma \cdot \Delta x_{ij}^{(t)} + c_1 \cdot (x_{kj}^{(t)} - x_{ij}^{(t)}) + c_2 \cdot \varepsilon(\overline{x}_j^{(t)} - x_{ij}^{(t)}), \tag{1}$$

$$x_{ij}^{(t+1)} = \begin{cases} x_{ij}^{(t)} + \triangle\, x_{ij}^{(t+1)}, & if\ p_i^{(t)} \le p_i^{(L)}, \\ x_{ij}^{(t)}, & otherwise. \end{cases} \quad (2)$$

where, $\boldsymbol{x}_i^{(t)} = (x_{i1}^{(t)}, x_{i2}^{(t)}, ..., x_{iD}^{(t)})$, $1 \le i < N$ denotes the position vector of the ith particle at generation t. N and D represent the population size and the dimensionality of the problem, respectively. γ is the inertia weight. $\Delta\boldsymbol{x}_i^{(t)} = (\Delta x_{i1}^{(t)}, \Delta x_{i2}^{(t)}, ..., \Delta x_{iD}^{(t)})$ denotes the behavior correction vector acting similarly to the velocity correction vector in the PSO. j denotes the jth decision variable for the ith particle, and $P_i^{(L)}$ indicates the learning probability of the ith particle, γ, c_1, c_2 and $p_i^{(t)}$ are three uniformly distributed random numbers in [0,1] respectively. $x_{kj}^{(t)}$ represents the jth element of the kth exemplar particle with better fitness over the ith particle, and $\overline{x}_j^{(t)} = \left(\sum_{i=1}^{N} x_{ij}^{(t)}\right)/N$ denotes the mean value of the iterative swarm in the jth dimension. In Eq. (1), ε is the social influence factor for controlling the effect of $\overline{x}_j^{(t)}$ on behavior learning. In this work, P_i^L and ε are set to 1 and 0, respectively [29].

Radial basis function

DDCSAEA constructs an RBF model to approximate the projected neighborhood landscape of the iterative swarm in the feature subspace during the surrogate-assisted feature subspace optimization phase, and to estimate the fitness of the new candidate samples in the feature subspace. RBF, as a single hidden-layer feed-forward neural network, is more effective at approximating target problems with different orders of nonlinearities and varying landscape characteristics than GP, PR and SVR, and has the advantage of being less sensitive to the training sample size and problem scale [30–32].

Formally, an RBF model can be obtained by interpolating the N pairs of training data $(\mathbf{x}_1, f(\mathbf{x}_1))$, $(\mathbf{x}_2, f(\mathbf{x}_2))$..., $(\mathbf{x}_N, f(\mathbf{x}_N))$, $\mathbf{x}_i \in \mathbf{R}^d$, $f(\mathbf{x}_i) \in R$, $i = 1, 2, ..., N$, according to Eq. (3) [33].

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}), \quad (3)$$

where $\|\cdot\|$ and $\varphi(\cdot)$ denote the Euclidean norm and kernel basis function, respectively. Commonly used kernel basis functions include cubic splines, thin-plate splines, gaussian, linear splines, and multi-quadrics splines. In this work, we opt for the thin-plate spline to construct the RBF to approximate the landscape of the feature subspace due to its excellent smoothing performance [34]. In Eq. (3), $\alpha_i \in R$ represents the interpolation weight of kernel basis function over $\mathbf{x}_i$. $p(\mathbf{x})$ indicates a linear polynomial in d variables that meets $\sum_{i=1}^{N} \alpha_i p(\mathbf{x}_i) = 0$. The hyperparameters in Eq. (3) can be derived from the following system of equations.

$$\begin{pmatrix} \boldsymbol{\Phi} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \end{pmatrix} \quad (4)$$

where $\boldsymbol{\Phi} \in \mathbf{R}^{N\times N}$ is the kernel function matrix filled with $\boldsymbol{\Phi}_{ij} := \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|),i,\ j = 1, 2, ..., N.\boldsymbol{\alpha}= (\alpha_1, \alpha_2, ..., \alpha_N)^{\mathrm{T}} \in \mathbf{R}^N$ denotes the weight coefficient vector. $\mathbf{P} \in \mathbf{R}^{N\times(d+1)}$ collects the values of the primary functions of the linear polynomial $p(\boldsymbol{x})$ at the interpolated sample points, and the vector $\mathbf{c}=(c_1, c_2, ..., c_{d+1})^T \in \mathbf{R}^{d+1}$ gathers the coefficients of the linear polynomial $p(\boldsymbol{x})$. $\mathbf{F} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), ..., f(\boldsymbol{x}_N))^T \in \mathbf{R}^N$ is the vector of fitness for the interpolated samples. Here the necessary condition for the coefficient matrix in Eq. (4) to be non-singular is that all training samples are affinely independent [35].

Principal component analysis for feature reduction

DDCSAEA takes PCA to implement the dimension reduction for producing a low-dimensional feature subspace by extracting as much principal feature information as possible from the original high-dimensional space, thereby effectively controlling the trade-off of the problem complexity and surrogate modeling complexity and improving the efficiency and accuracy of surrogate-assisted optimization. PCA is an unsupervised feature learning technique often used for dimension reduction of high-dimensional data [36]. The principle of PCA is to generate a set of uncorrelated low-dimensional feature vectors by performing a singular value decomposition on the covariance matrix of the centralized high-dimensional training samples. The new samples are then linearly projected into the low-dimensional feature space.

DDCSAEA randomly chooses M samples $\mathbf{X}^{(rand)} = (\mathbf{X}_{\mathbf{1}}^{(rand)}, \mathbf{X}_{\mathbf{2}}^{(rand)}, ..., \mathbf{X}_{\mathbf{M}}^{(rand)})$ from the database in the original feature space as the training samples for PCA projection modeling. Here using a random sample set of the database for the PCA training, on the one hand, allows the iterative population to follow the potential exploration and exploitation directions with equal probability as the optimization proceeds, thus weakening the biased search due to the biased distribution of the evolutionary samples. On the other hand, in this way, the optimal neighborhood prior to the target problem, especially the prior knowledge of the unexplored optimal regions, can be enriched following the low-dimensional subspace search on the optimal domain covered by the iterative population. The basic steps for constructing the low-dimension feature subspace by PCA on the random sample set $\mathbf{X}^{(rand)}$ in D-dimensional feature space are as follows.

(1) Centralize $\mathbf{X}^{(\boldsymbol{rand})} = (\mathbf{x}_i^{(\boldsymbol{rand})}, \mathbf{x}_\mathbf{2}^{(\boldsymbol{rand})}, ..., \mathbf{x}_\mathbf{M}^{(\boldsymbol{rand})})$ to yield $\mathbf{X}_c^{(\boldsymbol{rand})} = (\mathbf{x}_{c1}^{(\boldsymbol{rand})}, \mathbf{x}_{c2}^{(\boldsymbol{rand})}, ..., \mathbf{x}_{cM}^{(\boldsymbol{rand})})$, where $\mathbf{x}_i^{(rand)} = (x_{i1}^{(rand)}, x_{i2}^{(rand)}, ..., x_{iD}^{(rand)})^T$, $\mathbf{x}_{ci}^{(\boldsymbol{rand})} = \mathbf{x}_i^{(\boldsymbol{rand})} - \mathbf{x}_c, i = 1, 2, ..., M, \mathbf{x}_c = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i^{(\boldsymbol{rand})}$;

(2) Calculate the covariance matrix $\mathbf{R} = \mathbf{X}_c^{(\boldsymbol{rand})}(\mathbf{X}_c^{(\boldsymbol{rand})})^T$ of $\mathbf{X}_c^{(\boldsymbol{rand})}$ and perform singular value decomposition on $\mathbf{R}$ to obtain $\mathbf{R} = \mathbf{UDV}^\mathrm{T}$;

(3) Take the first D_s $(D_s < D)$ columns of the matrix $\mathbf{U}$ to comprise the low-dimensional feature vector set $\mathbf{U}_{map} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{Ds})$, and construct the D_s dimensional feature subspace.

This work sets the feature subspace size $D_s = 10$ to balance the optimization performance and feature information loss of the DDCSAEA. The sensitivity analysis on the parameter D_s is detailed in "Sensitivity analysis". Based on the PCA model, the projection vector $\mathbf{X}_{new}^{(sub)}$ in the feature subspace of the newly added sample $\mathbf{X}_{new}$ in the original high-dimensional feature space can be derived from $\mathbf{X}_{new}^{(sub)} = (\mathbf{U}_{map})^T(\mathbf{X}_{new} - \mathbf{X}_c)$.

Autoencoder for feature reconstruction

DDCSAEA selects the candidate solutions obtained from the low-dimensional feature subspace as input data to train an autoencoder [37, 38], which is then adopted to reconstruct these candidate solutions into the original high-dimensional feature space for infill-sampling. The autoencoder is also an unsupervised feature learning technique depending on a backpropagation algorithm and optimization methods [39, 40]. As an important feature learning technique, autoencoder has been widely used in areas such as image classification and pattern recognition [41]. The autoencoder is essentially made up of a binary symmetric structure with an encoder and a decoder, as shown in Fig. 1a.

Given p input data $\mathbf{X}^{input} = (\mathbf{X}_1^{input}, \mathbf{X}_2^{input}, ..., \mathbf{X}_p^{input})$, the autoencoder first encodes $\mathbf{X}^{input}$ to map it to the hidden layer's embedding space: $Encoder(\mathbf{X}^{input}) \rightarrow \mathbf{X}^{embedding}$, and then reconstructs $\mathbf{X}^{embedding}$ at the output end through decoding to map it back to the original input space: $Decoder(\mathbf{X}^{embedding}) \rightarrow \mathbf{X}^{\mathrm{output}}$. Mathematically, Eq. (5) formulates the encoding process of $\mathbf{X}^{input}$ from the input space to the embedding space of the hidden layer. The decoding process of $\mathbf{X}^{embedding}$ from the hidden layer's embedding space to the output space is formulated in Eq. (6).

$$\mathbf{X}_j^{embedding} = \sigma(\mathbf{W}\mathbf{X}_j^{input} + \mathbf{b}) \tag{5}$$

$$\mathbf{X}_i^{output} = \sigma(\mathbf{W}'\mathbf{X}_j^{embedding} + \mathbf{b}) \tag{6}$$

where $\mathbf{X}_j^{input}$ denotes the j^{th} input data, $\mathbf{X}_j^{embedding}$ and $\mathbf{X}_i^{output}$ represent the encoded vector and the decoded vector of $\mathbf{X}_j^{input}$, respectively. $\sigma(\cdot)$ represents the codec function and the Sigmoid function is chosen in this work for nonlinear codec transformation. $\mathbf{W} \in \mathbb{R}^{D_s \times D}$ and $\mathbf{b} \in D_s$ are the weight matrix and bias vector of the encoding process, respectively. $\mathbf{W}' \in \mathbb{R}^{D \times D_s}$ and $\mathbf{b}' \in D$ are the weight matrix and bias vector of the decoding process, respectively. The weights and bias units in the codec process can be derived by minimizing the reconstruction error $L(w, b)$ of the input data $\mathbf{X}^{input}$ and the output data $\mathbf{X}^{output}$, as shown in Eq. (7).

$$L(w, b) = \sum_{i=1}^{p} \left\| \mathbf{X}_i^{input} - \mathbf{X}_i^{output} \right\|^2 \tag{7}$$

Unlike the reconstruction mode of linear PCA, the autoencoder learns the bilateral non-linear encoding and decoding transformations from input data to output data by minimizing the reconstruction error, so that it can maintain the consistency of data information between the input and output layers as much as possible, resulting in a strong generalization performance [42]. The autoencoder can achieve feature downscaling and feature upscaling of the input data during its encoding process by scaling the number of neuron nodes in the hidden layer [43]. Therefore, DDCSAEA uses an autoencoder to encode the candidate solutions of the low-dimensional feature subspace to expand their dimensionality for reconstructing the corresponding samples of the original high-dimensional feature space. In this way, the optimal structural property information about the problem fitness landscape contained in the candidate solutions of the low-dimensional feature subspace allows being transferred. To balance the complexity of autoencoder training and the efficiency of DDASAEA, the training epoch for the Autoencoder is set as $epoch = 20$. The PCA feature reduction and autoencoder feature reconstruction are shown in Fig. 1b.

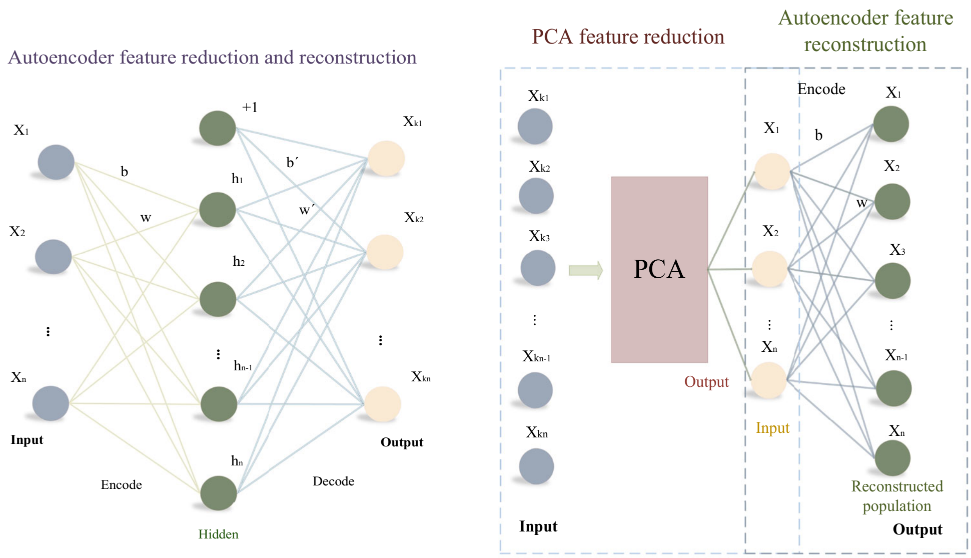
Dual-drive collaboration surrogate-assisted evolutionary algorithm by coupling feature reduction and reconstruction

Outline of DDCSAEA

Algorithm 1 gives the pseudocode of the DDCSAEA. The schema of DDCSAEA is mainly structured in two modules. One features a PCA-driven feature reduction and an RBF-assisted evolutionary sampling in the feature subspace. The other accompanies an autoencoder-driven feature reconstruction and nearest neighbor principle-assisted infill sampling in the original high-dimensional space. The two modules execute in tandem to realize the transfer and communication nearest neighbors surrounding the optimal solution attained in the feature subspace optimization to fully exploit the optimal a priori of the solution space.

Algorithm 1: Pseudocode of DDCSAEA

```
Input: Scaling factor $F$, Crossover rate $CR$, Population size $N$, Maximum evaluation threshold
    $MaxFes$, Problem dimension $D$, the training sample size $NS$ for Autoencoder modeling.
Output: Global best solution $(\mathbf{Gbest}, F(\mathbf{Gbest}))$.
Begin
Initialize the population $\mathbf{P}^{(t)} = (\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N)$ by LHD, and compute their real fitness $F(\mathbf{P}^{(t)})$,
    determine the current global best solution $(\mathbf{Gbest}, F(\mathbf{Gbest}))$. Archive them into the database
    DB, and set $t = 0$, $Fes = N$.
while $Fes<MaxFes$
    Sort DB in ascending order of fitness, and set $\mathbf{DB} = \mathbf{DB}_{sort}$;
    Compute $\mathbf{Gbest}_{RBF}$ by Algorithm 2;
    Obtain the reconstructed solutions $\mathbf{S}^{(recons)} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{NS})$ by Algorithm 3;
    Determine $q$ candidate solutions $\mathbf{nbest} = (\mathbf{ns}_1, \mathbf{ns}_2, ..., \mathbf{ns}_q)$ from $\mathbf{S}^{recons}$ in the original high-
    dimensional feature space by Algorithm 4;
    Reevaluate $F(\mathbf{nbest})$, and archive them into the database DB, set $Fes = Fes + q$;
    for $i = 1, 2, ..., p$ do
        if $F(\mathbf{ns}_i) < F(\mathbf{Gbest})$ then
            $\mathbf{Gbest} = \mathbf{ns}_i$;
            $F(\mathbf{Gbest}) = F(\mathbf{ns}_i)$;
        end if
    end for
    update the population $\mathbf{P}^{(t)}$ using the top-ranking samples in the DB, and set $t = t + 1$;
end while
end
```



(a) (b)

Fig. 1 Schematic diagrams for feature reduction and reconstruction: **a** feature reduction and reconstruction by a single autoencoder; **b** feature reduction and reconstruction by PCA and an autoencoder, respectively

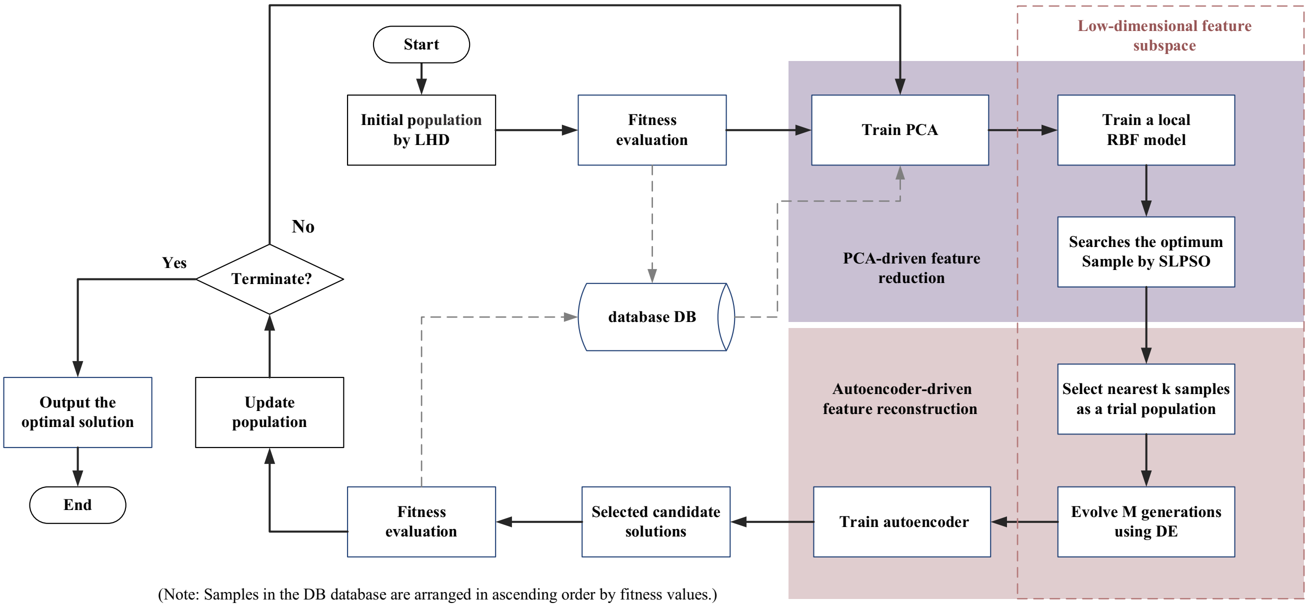

Start
Initial population by LHD
Fitness evaluation
Train PCA
Low-dimensional feature subspace
Train a local RBF model
Searches the optimum Sample by SLPSO
PCA-driven feature reduction
No
Yes
Terminate?
database DB
Output the optimal solution
Update population
Autoencoder-driven feature reconstruction
Select nearest k samples as a trial population
End
Fitness evaluation
Selected candidate solutions
Train autoencoder
Evolve M generations using DE
(Note: Samples in the DB database are arranged in ascending order by fitness values.)

Fig. 2 The flowchart of DDCSAEA

Figure 2 presents the flowchart of DDCSAEA, wherein the dashed line indicates the data flow direction. As shown in Fig. 2, DDCSAEA first uses Latin hypercube sampling (LHS) [44] to initialize the iterative swarm with its fitness calculated by the real objective function. Then the real-evaluated samples are stored in the database DB, followed by an ascending ranking of the DB samples based on fitness. Afterward, the PCA-driven feature reduction on the original high-dimensional space is performed to construct a low-dimensional feature subspace. Here, a certain number of random samples of the database is utilized to construct and update the PCA to reduce the original high-dimensional problem space into a low-dimensional feature subspace. The projection of the random sample set in the original problem space is selected to train a local RBF model to approximate the landscape of the feature subspace, and the optimum $\mathbf{Gbest}_{RBF}$ of the RBF model in the projection domain of the iterative swarm in the feature subspace is located via the SLPSO. Thereafter, k nearest neighbors to $\mathbf{Gbest}_{RBF}$ are chosen to constitute a trial population according to the Euclidean distance criterion. The trial population then evolves M generations with $DE/rand/1$ mutation operator to generate $K \times M$ candidate solutions. Subsequently, the autoencoder-driven feature reconstruction on these candidate solutions is implemented to reconstruct these candidate solutions into the original high-dimensional space in a one-by-one manner. Here a certain number of randomly selected candidate solutions is used to train the autoencoder. Finally, the Euclidean distances between the current global best solution and reconstructed candidate solutions are computed, and the solutions with the smallest distance are then chosen for real evaluation. During the iteration, the newly evaluated solutions are archived in DB to update the database and the global optimum. Repeat the above procedure until the termination condition is reached, then output the optimal solution.

PCA-driven feature reduction and surrogate-assisted feature subspace optimization

To alleviate the high complexity of surrogate modeling in high-dimensional feature space and improve the prediction accuracy, DDCSAEA iteratively trains a PCA model to perform feature reduction on the original high-dimensional feature space to extract principal features for constructing a simplified low-dimensional feature subspace. A local RBF model is then trained by the projections in the feature subspace of a certain number of random samples in the database to approximate the landscape of the feature subspace. To further manipulate the training complexity of RBF in the feature subspace, the $2D_s + 1$[45] database samples are chosen for training the RBF model, where D_s denotes the dimension of the feature subspace. Moreover, to fully exploit a priori knowledge contained in the principal features, an RBF-assisted SLPSO search is carried out to find the best solution $\mathbf{Gbest}_{RBF}$ of the RBF model in the area covered by the training samples. Note that other EAs and local search methods can also be used as the local optimizer. Algorithm 2 provides a pseudo-code for the surrogate model-assisted feature subspace optimization based on the PCA feature reduction.

```
Algorithm 2: Surrogate-assisted feature subspace optimization based on PCA
Input: Current population $\mathbf{P}^{(t)}$, reduced dimension $D_s$, population size $N$, number of training
    samples $M$, Database after sorting DB.
Output: $\mathbf{Gbest}_{RBF}$.
Begin
    Randomly select $M$ samples $\mathbf{X}^{(rand)}$ from DB;
    Compute $\mathbf{x}_c = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i^{(rand)}$;
    Centralize $\mathbf{X}^{(rand)} = (\mathbf{x}_i^{(rand)}, \mathbf{x}_2^{(rand)}, ..., \mathbf{x}_\mathbf{M}^{(rand)})$ to yield $\mathbf{X}_c^{(rand)} = (\mathbf{x}_{c1}^{(rand)}, \mathbf{x}_{c2}^{(rand)}, ..., \mathbf{x}_{cM}^{(rand)})$;
    Calculate the covariance matrix $\mathbf{R} = \mathbf{X}_c^{(rand)}(\mathbf{X}_c^{(rand)})^T$;
    Perform singular value decomposition on $\mathbf{R}$ to obtain $\mathbf{R} = \mathbf{UDV}^\mathrm{T}$;
    Take the first $D_s$ columns of the matrix $\mathbf{U}$ to obtain the low-dimensional feature vector set
        $\mathbf{U}_{map} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{Ds})$;
    Randomly choose $2D_s + 1$ samples of DB to construct the RBF model;
    Find the global best solution $\mathbf{Gbest}_{RBF}$ of the RBF model by invoking the SLPSO optimizer;
end
```

Autoencoder-driven feature reconstruction

After completing the surrogate-assisted feature subspace optimization, DDCSAEA further collects the candidate samples $\mathbf{V}^{(candi)}$ after deduplication in the feature subspace to train an autoencoder for feature reconstruction. Algorithm 3

```
Algorithm 3: Autoencoder-driven feature reconstruction
Input: Scaling factor $F$, crossover rate $CR$, $\mathbf{Gbest}_{RBF}$, the training sample size for AC modeling
    $NS$, population size $N$, mapped population $\mathbf{PoP}^{(t)} = (\mathbf{pop}_1, \mathbf{pop}_2, ..., \mathbf{pop}_N)$.
Output: $\mathbf{S}^{(recons)} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{dn})$
Begin
for $i = 1, 2, ..., N$ do
    Calculate $dis_i = \|\mathbf{pop}_i - \mathbf{Gbest}_{RBF}\|_2$;
end for
Set $Dis = (dis_1, dis_2, ..., dis_N)$, and sort $Dis$ in ascending order;
Take the first K samples with the smallest distance to $\mathbf{Gbest}_{RBF}$ from $\mathbf{PoP}^{(t)}$ to comprise the
    trial population $\mathbf{P}^{(trial)}$;
Evolve $\mathbf{P}^{(trial)}$ $M$ generations by DE/rand/1 mutation and binomial crossover operators to acquire
    the candidate solution set $\mathbf{V}^{(candi)} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{dn})$ with no duplicate;
if $dn < NS$ do
    Select the first $NS - dn$ top-ranking samples $\mathbf{pop}_{best}$ from $\mathbf{PoP}^{(t)}$, set
        $\mathbf{V}^{(candi)} = [\mathbf{V}^{(candi)}; \mathbf{pop}_{best}]$, and use $\mathbf{V}^{(candi)}$ to build the AC model;
else
    Randomly select NS samples $\mathbf{V}_{rand}$ from $\mathbf{V}^{(candi)}$, set $\mathbf{V}^{(candi)} = \mathbf{V}_{rand}$;
end if
Build the AC model using $\mathbf{V}^{(candi)}$;
Encode $\mathbf{V}^{(candi)}$ to obtain the corresponding reconstructed solution $\mathbf{S}^{(recons)} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{NS})$ in
    the original high-dimension feature space;
end
```

provides the pseudocode for autoencoder-driven feature reconstruction, where $\mathbf{S}^{(recons)}$ indicates the set of reconstructed solutions in the original problem space $\mathbb{R}^{D}$ by the autoencoder on the associated candidate solution set in the feature subspace $\mathbb{R}^{D_s}$.

To diversify the candidate solutions for the autoencoder modeling and the subsequent infill-sampling, the nearest neighbors to $\mathbf{Gbest}_{RBF}$ further undergo several generations of differential mutation and crossover operations to generate a set of trial solutions. In this way, it is possible to assure the intra-domain transfer of information about the optimal structure of the feature subspace on the one hand. On the other hand, it can effectively enrich a priori landscape knowledge over the optimal region of the feature subspace and provide enough samples for subsequent feature reconstruction. To be more specific, the Euclidean distances between the optimal solution $\mathbf{Gbest}_{RBF}$ and the individuals of the iterative population $\mathbf{PoP}^{(t)}$ is first calculated. Here $\mathbf{PoP}^{(t)}$ represents the corresponding projected population of feature subspace $\mathbb{R}^{D_s}$ by projecting the iterative population $\mathbf{P}^{(t)}$ of the original problem space $\mathbb{R}^{D}$ with the PCA model. K individuals with the smallest distances are then figured out to comprise a trial population $\mathbf{P}^{(trial)}$. After that, $\mathbf{P}^{(trial)}$ performs M generations of differential mutation and binomial crossover operations to generate the candidate solution set $\mathbf{V}^{(candi)}$. In this work, $DE/rand/1$ mutation strategy as shown in Eq. (8) is adopted, where x_{r1}, x_{r2} and x_{r2} represent three mutually exclusive individuals in $\mathbf{P}^{(trial)}$ and F is the scale factor taking values in the interval [0.4, 1]

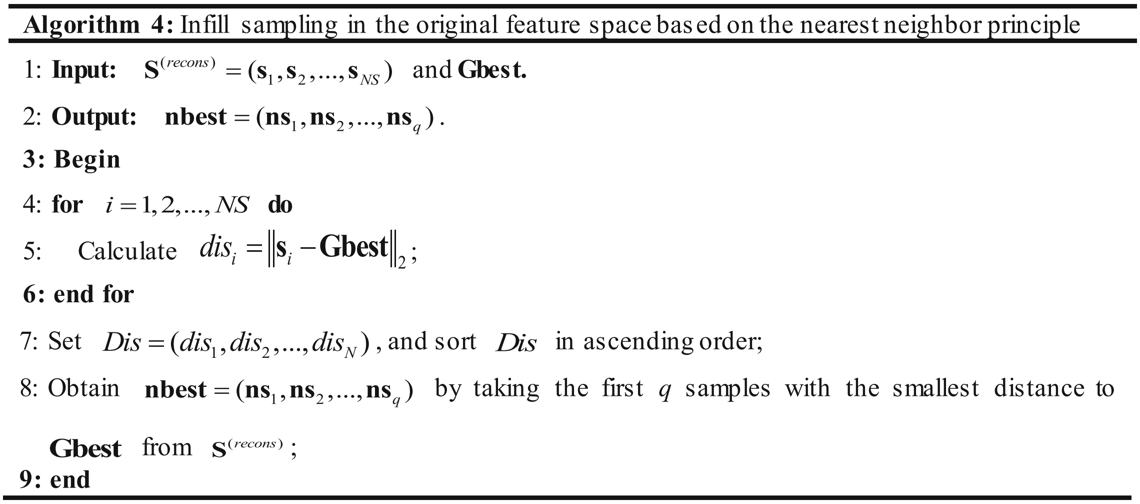
$$\upsilon_i = x_{r1} + F(x_{r2} - x_{r3}) \tag{8}$$

For a balance between the training efficiency and prediction accuracy of the autoencoder model, its training set size is set to NS. If the number of candidate solutions in the feature subspace exceeds NS, NS candidate solutions are randomly picked from the candidate set to train the autoencoder model; Otherwise, a fraction of individuals in the trial population $\mathbf{P}^{(trial)}$ are chosen in the order of fitness priority to compensate for the training set. Note that the training set of the autoencoder model undergoes feature reconstruction to acquire the candidate solution set of the original high-dimensional feature space for subsequent infill sampling.

It is conceivable that using the autoencoder model to non-linearly transfer the structural prior information concerning the optimal region of the feature subspace to the original problem space at the expense of losing some feature information contributes to the dynamic regulation of the structural prior information of the original problem space. Meanwhile, as the newly added samples aggregate within the neighborhood of the best solution, the consistency regarding the optimal structure attributes between the feature subspace $\mathbb{R}^{D_s}$ and the original problem space $\mathbb{R}^{D}$ is possible to strike a good balance via the autoencoder-driven feature reconstruction.

Infill-sampling based on the nearest neighbor principle

After mapping the candidate solutions in the feature subspace back into the original problem space via the autoencoder model, q samples $\mathbf{nbest} = (\mathbf{ns}_1, \mathbf{ns}_2, ..., \mathbf{ns}_q)$ nearest to the current global best solution $\mathbf{Gbest}$ are picked from the candidate solution set $\mathbf{S}^{(recons)}$ for real evaluation. Algorithm 4 provides the pseudocode of the infill-sampling strategy. Here, taking the nearest neighbors of the best solution for reevaluation and infill-sampling is capable of increasing the sample density of the neighborhood the optimal solution locates, thus enriching the landscape prior to steering the search to rapidly reach the optimum.



Algorithm 4: Infill sampling in the original feature space based on the nearest neighbor principle

```
Input: $\mathbf{S}^{(recons)} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{NS})$ and Gbest.
Output: $\mathbf{nbest} = (\mathbf{ns}_1, \mathbf{ns}_2, ..., \mathbf{ns}_q)$.
Begin
for $i = 1, 2, ..., NS$ do
    Calculate $dis_i = \|\mathbf{s}_i - \mathbf{Gbest}\|_2$;
end for
Set $Dis = (dis_1, dis_2, ..., dis_N)$, and sort $Dis$ in ascending order;
Obtain $\mathbf{nbest} = (\mathbf{ns}_1, \mathbf{ns}_2, ..., \mathbf{ns}_q)$ by taking the first $q$ samples with the smallest distance to
    Gbest from $\mathbf{S}^{(recons)}$;
end
```

Table 1 Basic characteristics of selected benchmark problems

No	Function	Dimension	Variable domain	Characteristics	Global optimum
F1	Ellipsoid	30, 50, 100, 200	$[-5, 5]^D$	Unimodal	0
F2	Rosenbrock	30, 50, 100, 200	$[-2, 2]^D$	Multimodal with narrow valley	0
F3	Ackley	30, 50, 100, 200	$[-32, 32]^D$	Multimodal	0
F4	Griewank	30, 50, 100, 200	$[-600, 600]^D$	Multimodal	0
F5	Rastrigin	30, 50, 100, 200	$[-5, 5]^D$	Multimodal	0
F6	Shifted Rotated Rastrigin (F_{10} in [49])	30, 50, 100, 200	$[-5, 5]^D$	Very complicated multimodal	− 330
F7	Rotated Hybrid composition function with a Narrow Basin for the Global Optimum (F_{19} in [49])	30, 50, 100, 200	$[-5, 5]^D$	Very complicated multimodal	10

Empirical study

To validate the effectiveness of the proposed DDCSAEA, we test it over seven widely used benchmark problems featured by different fitness landscapes at four-dimensional scales, i.e., 30, 50, 100, and 200 dimensions. Its performance is compared with eight state-of-the-art algorithms, including SHPSO [9], TL-SSLPSO [46], SAMSO [23], DESO [47], TS-DDEO [48], TASEA [22], SADE-AMSS [25] and SAEO [28], to examine its optimization efficiency. Table 1 lists the basic characteristics of the selected benchmark problems that cover the single-peaked landscape with zero-point as optimum, multi-peaked landscapes with zero-point as optimum, and complex asymmetric multi-peaked landscapes with non-zero-point as optimum.

Parameter settings

In the following experiments, the population size for DDCSAEA is set to 50. $D_s = 10$ principal features are extracted via PCA for structuring the feature subspace $\mathbb{R}^{D_s}$. The scale factor F and the crossover probability CR are both set to 0.8, and $K = 5$ nearest neighbors to **Gbest** are picked to comprise the trial population $\mathbf{P}^{(trial)}$. $NS = 80$ training samples are chosen for the autoencoder modeling. A sensitivity analysis concerning the parameters K and D_s on the performance of DDCSAEA is detailed in the subsequent "Sensitivity analysis". To control the computational budget, $q = 2$ solutions are chosen for real evaluation at each iteration. In addition, the parameter configurations for the compared algorithms SHPSO, TL-SSLPSO, SAMSO, DESO, TS-DDEO, TASEA, SADE-AMSS and SAEO follow the same settings recommended in the relevant literature. All algorithms involved in the experiments are implemented on a desktop computer with an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz. 20 independent runs are assigned to each contestant, and the maximum number of real objective function evaluations $MaxFes = 1000$ is used to trigger the termination condition.

Behavior analysis of the DDCSAEA

DDCSAEA sequentially couples the PCA model and autoencoder model for feature reduction and feature reconstruction of the original high-dimensional problem space. To test its effectiveness, in this section, we first perform a sensitivity analysis of parameter D_s to gain some insights into its impacts on the performance of DDCSAEA. Then, we draw a comparison on the efficiency of feature reduction and reconstruction between DDCSAEA coupling with PCA and autoencoder versus DDCSAEA solely assembling the autoencoder or the PCA.

Sensitivity analysis

Table 2 records the statistical results of DDCSAEA with $D_s = 5, 10, 30$ on 50-, 100-, and 200-dimensional benchmark problems over 20 independent runs, respectively, including the mean and standard deviation of the obtained best solutions as well as the average time cost (TC). In Table 2, the best result on each test instance is bolded and the suboptimal result is highlighted with shadow. As shown in Table 2, DDCSAEA's performance improves as the feature subspace scale grows, but so does its time complexity. More specifically, DDCSAEA possesses a low time complexity with $D_s = 5$, but it is hard to locate a better solution to the selected problems. When the scale of D_s reaches $D_s = 30$, DDCSAEA performs the best on a majority of the selected problems. However, its time complexity is higher among the comparative cases. In contrast, for DDCSAEA with $D_s = 10$, its final results are slightly worse than that of DDCSAEA with $D_s = 30$, but its time complexity has received a great boost. Meanwhile, compared to DDCSAEA with $D_s = 5$, DDCSAEA taking $D_s = 10$ gets better performance, and both have a relatively lower time complexity

Table 2 The statistical results of DDCSAEA with different D_s

No	D	$D_s = 5$		$D_s = 10$		$D_s = 30$	
		Mean (Std.)	TC	Mean (Std.)	TC	Mean (Std.)	TC
F1	50	9.54e − 02 (1.46e − 02)	**9.64e + 01**	1.46e − 02 (1.79e − 02)	1.28e + 02	**5.14e − 03 (6.05e − 03)**	3.88e + 02
	100	1.46e − 01 (2.23e − 02)	**1.02e + 02**	**2.15e − 02 (4.05e − 02)**	1.36e + 02	8.14e − 02 (1.00e − 01)	4.04e + 02
	200	5.33e − 01 (3.01e − 01)	**1.05e + 02**	1.98e − 01 (5.94e − 02)	1.56e + 02	**3.99e − 02 (6.94e − 02)**	4.28e + 02
F2	50	4.85e + 01 (5.40e − 03)	**9.43e + 01**	4.85e + 01 (6.58e − 03)	1.25e + 02	**4.82e + 01 (1.94e − 02)**	3.86e + 02
	100	9.80e + 01 (1.63e − 02)	**9.90e + 01**	9.80e + 01 (1.19e − 02)	1.33e + 02	**9.80e + 01 (6.66e − 03)**	3.91e + 02
	200	**1.97e + 02 (4.29e − 02)**	**1.09e + 02**	1.97e + 02 (7.29e − 02)	1.49e + 02	1.97e + 02 (1.06e − 01)	3.79e + 02
F3	50	3.36e − 02 (1.76e − 02)	**9.25e + 01**	1.74e − 02 (1.45e − 02)	1.13e + 02	**3.65e − 03 (2.75e − 02)**	3.04e + 02
	100	5.20e − 02 (2.36e − 02)	**9.76e + 01**	5.14e − 02 (1.95e − 02)	1.17e + 02	**1.47e − 02 (1.24e − 02)**	3.09e + 02
	200	7.99e − 02 (3.63e − 02)	**1.07e + 02**	4.42e − 02 (3.36e − 02)	1.50e + 02	**2.17e − 02 (1.63e − 02)**	2.83e + 02
F4	50	1.04e − 04 (1.08e − 04)	**9.19e + 01**	**9.41e − 05 (9.36e − 05)**	1.12e + 02	3.36e − 04 (4.94e − 04)	3.26e + 02
	100	4.25e − 04 (1.94e − 04)	**9.73e + 01**	3.45e − 04 (1.53e − 04)	1.21e + 02	**2.23e − 04 (3.02e − 03)**	3.43e + 02
	200	8.56e − 04 (3.16e − 04)	**1.17e + 02**	**7.42e − 04 (2.66e − 04)**	1.62e + 02	1.63e − 03 (2.76e − 03)	3.26e + 02
F5	50	7.19e − 02 (9.24e − 02)	**9.16e + 01**	9.30e − 02 (1.31e − 01)	1.09e + 02	**6.24e − 02 (7.88e − 02)**	2.82e + 02
	100	3.33e − 01 (6.11e − 01)	**9.96e + 01**	9.51e − 02 (1.88e − 01)	1.15e + 02	**9.45e − 02 (1.93e − 01)**	3.66e + 02
	200	6.10e − 01 (1.20e + 00)	**1.09e + 02**	**1.13e − 01 (1.49e − 01)**	1.51e + 02	1.63e − 01 (2.38e − 01)	3.39e + 02
F6	50	1.03e + 03 (1.26e + 01)	**1.02e + 02**	1.03e + 03 (1.67e + 01)	1.11e + 02	**1.02e + 03 (1.28e + 01)**	2.64e + 02
	100	1.99e + 03 (3.90e + 01)	**1.16e + 02**	2.00e + 03 (3.52e + 01)	1.26e + 02	**1.98e + 03 (5.26e + 01)**	3.75e + 02
	200	4.39e + 03 (1.03e + 02)	**1.59e + 02**	4.37e + 03 (1.06e + 02)	2.04e + 02	**4.37e + 03 (8.81e + 01)**	4.69e + 02
F7	50	9.10e + 02 (7.44e − 02)	**1.01e + 02**	9.10e + 02 (1.80e − 02)	1.17e + 02	**9.10e + 02 (1.58e − 02)**	2.92e + 02
	100	9.10e + 02 (5.53e − 02)	**1.96e + 02**	**9.10e + 02 (2.04e − 02)**	1.95e + 02	9.10e + 02 (2.88e − 02)	4.31e + 02
	200	9.10e + 02 (2.34e − 02)	**5.90e + 02**	9.10e + 02 (9.92e − 03)	6.24e + 02	**9.10e + 02 (8.43e − 03)**	9.22e + 02

against the case of $D_s = 30$. Therefore, to strike a good trade-off of the computational complexity and convergence performance of the DDCSAEA, $D_s = 10$ is configured to regulate the feature subspace scale.

Comparison results of DDCSAEA featured by a hybrid or single feature reduction and reconstruction technique

For simplicity, we denote the variant of DDCSAEA that ensembles a single autoencoder for feature reduction and reconstruction as DDCSAEA-AE, and name another DDCSAEA variant as DDCSAEA-PTP that solely assembles a PCA for feature reduction and reconstruction. Table 3 presents the statistical results of these three contestants for solving 50-, 100-, and 200-dimensional benchmark problems. The pairwise Wilcoxon rank sum test on the final results at 95% confidence level is also computed, where “ + ”, “–” and “≈” indicate DDCSAEA performs significantly better than, significantly worse than or equivalent to the compared algorithm, respectively, in terms of the final solutions. As shown in Table 3, for the selected benchmark problems, DDCSAEA significantly outperforms DDCSAEA-AE and DDCSAEA-PTP in obtaining the best solutions for at least 18 test instances, slightly underperforms DDCSAEA-PTP and DDCSAEA-AE on two test instances, and performs comparably to DDCSAEA-AE on one instance. To be more specific, DDCSAEA can obtain significantly better results than DDCSAEA-PTP and DDCSAEA-AE over 1000 real fitness function evaluations for all unimodal and multimodal test problems except F6, exhibiting good robustness over the problem scale. For F6 featured a complex multimodal fitness landscape, DDCSAEA performs worse than DDCSAEA-PTP and DDCSAEA-AE in the cases of 50 and 100 dimensions. However, as the scale and complexity of the problem grow, the performance of DDCSAEA coupled with PCA and autoencoder improves. Meanwhile, there shows no significant difference between DDCSAEA and DDCSAEA-AE in the solution quality on the 200-dimensional F6, on which the best solution obtained by DDCSAEA is slightly better than that of DDCSAEA-AE. From Table 3, we can conclude that in contrast to directly encoding and decoding the original high-dimensional problem space via a single autoencoder or a single PCA, sequentially coupling PCA with autoencoder for feature reduction and reconstruction of

Table 3 Comparison results of DDCSAEA against DDCSAEA-PTP and DDCSAEA-AE on benchmarks

No	D	DDCSAEA-PTP	DDCSAEA-AE	DDCSAEA
		Mean (Std.)	Mean (Std.)	Mean (Std.)
F1	50	3.02e + 02 (2.32e + 02) +	1.83e + 01 (8.09e + 00) +	**1.46e – 02 (1.79e – 02)**
	100	8.16e + 02 (3.25e + 02) +	1.06e + 02 (3.01e + 01) +	**2.15e – 02 (4.05e – 02)**
	200	4.62e + 03 (3.54e + 03) +	4.12e + 02 (1.45e + 02) +	**1.98e – 01 (5.94e – 02)**
F2	50	2.92e + 02 (1.84e + 02) +	6.01e + 01 (5.67e + 00) +	**4.85e + 01 (6.58e – 03)**
	100	4.52e + 02 (1.42e + 02) +	1.27e + 02 (1.56e + 01) +	**9.80e + 01 (1.19e – 02)**
	200	8.20e + 02 (2.57e + 02) +	2.56e + 02 (2.77e + 01) +	**1.97e + 02 (7.29e – 02)**
F3	50	1.16e + 01 (2.58e + 00) +	6.32e + 00 (7.29e – 01) +	**1.74e – 02 (1.45e – 02)**
	100	1.11e + 01 (2.23e + 00) +	6.36e + 00 (6.80e – 01) +	**5.14e – 02 (1.95e – 02)**
	200	1.02e + 01 (1.69e + 00) +	5.97e + 00 (5.39e – 01) +	**4.42e – 02 (3.36e – 02)**
F4	50	4.11e + 01 (3.45e + 01) +	1.54e + 01 (4.36 e + 00) +	**9.41e – 05 (9.36e – 05)**
	100	8.38e + 01 (5.51e + 01) +	2.62e + 01 (7.52 e + 00) +	**3.45e – 04 (1.53e – 04)**
	200	1.30e + 02 (1.00e + 02) +	3.96e + 01 (1.24e + 01) +	**7.42e – 04 (2.66e – 04)**
F5	50	4.51e + 02 (3.52e + 01) +	1.58e + 02 (6.71e + 01) +	**9.30e – 02 (1.31e – 01)**
	100	9.27e + 02 (5.74e + 01) +	3.15e + 02 (8.25e + 01) +	**9.51e – 02 (1.88e – 01)**
	200	1.91e + 03 (7.76e + 01) +	6.97e + 02 (1.48e + 02) +	**1.13e – 01 (1.49e – 01)**
F6	50	**8.75e + 02 (7.29e + 01) –**	**8.80e + 02 (4.82e + 01) –**	1.03e + 03 (1.67e + 01)
	100	1.94e + 03 (8.79e + 01) –	**1.88e + 03 (4.50e + 01) –**	2.00e + 03 (3.52e + 01)
	200	4.73e + 03 (1.08e + 02) +	4.42e + 03 (9.17e + 01) ≈	**4.37e + 03 (1.06e + 02)**
F7	50	1.26e + 03 (7.94e + 01) +	9.57e + 02 (2.23e + 01) +	**9.10e + 02 (1.80e – 02)**
	100	1.24e + 03 (8.55e + 01) +	9.55e + 02 (1.65e + 01) +	**9.10e + 02 (2.04e – 02)**
	200	1.13e + 03 (4.74e + 01) +	9.45e + 02 (1.22e + 01) +	**9.10e + 02 (9.92e – 03)**
+ / ≈ / –		19 / 0 / 2	18 / 1 / 2	N / A

the target feature space can significantly improve the performance of DDCSAEA, demonstrating the effectiveness and superiority of this hybrid codec strategy.

Comparison results of DDCSAEA with different training samples for PCA

PCA tends to extract the principal components of feature space with larger variance by the training samples, thus using different training samples for PCA modeling differentiates the extracted principal components, due to the different distribution and quality of the training samples. This subsection explores the potential impact on the performance of DDCSAEA with different training samples for PCA modeling. Here, the DDCSAEA compares with its two variants, i.e., DDCSAEA-TB and DDCSAEA-TW. In terms of PCA modeling, DDCSAEA-TB chooses the best sample set of DB to be the training samples, while DDCSAEA-TW uses the worst sample set of DB. Table 4 shows the statistical results of these three competitors on the selected benchmark problems with the best results on each instance being highlighted. From Table 4, one can observe that DDCSAEA can obtain the best results than DDCSAEA-TB and DDCSAEA-TW on all the test instances except 100- and 200-dimensional F6, indicating the remarkable superiority of using a random sample set in the database for PCA modeling. In fact, as the optimization process progresses, the best samples in DB aggregate in the decision space exacerbating the difficulty of exploration with smaller variances in the principal direction, while the worst samples in DB deteriorate the exploitation. In contrast, using the random samples for PCA training is promising to promote diversity in search and help escape the shackles of local optimality by emphasizing exploration of the unexplored areas with random principal components.

Comparison results of DDCSAEA versus five advanced algorithms without feature reduction

To further investigate the computational efficiency of the proposed DDCSAEA, we carry out a comparison of DDCSAEA with five advanced algorithms without feature reduction, including SHPSO, TL-SSLPSO, SAMSO, DESO, and TS-DDEO. Table 5 shows the statistical results of these contestants on the selected benchmark problems. As shown in Table 5, in general, DDCSAEA performs significantly better than the five competitors for a majority of the selected test instances. According to the results derived from the pairwise

Table 4 Comparison results of DDCSAEA against DDCSAEA-TB and DDCSAEA-TW on the selected benchmarks

No	D	DDCSAEA-TB	DDCSAEA-TW	DDCSAEA
		Mean (Std.)	Mean (Std.)	Mean (Std.)
F1	50	3.50e − 01 (2.93e − 01) +	4.35e − 01 (6.09e − 01) +	**1.46e − 02 (1.79e − 02)**
	100	1.23e + 00 (1.22e + 00) +	1.94e + 00 (1.76e + 00) +	**2.15e − 02 (4.05e − 02)**
	200	5.87e + 00 (6.03e + 00) +	9.02e + 00 (8.75e + 00) +	**1.98e − 01 (5.94e − 02)**
F2	50	4.88e + 01 (3.99e − 01) +	4.87e + 01 (2.20e − 01) ≈	**4.85e + 01 (6.58e − 03)**
	100	9.94e + 01 (1.22 e + 00) +	9.89e + 01 (1.15e + 00) +	**9.80e + 01 (1.19e − 02)**
	200	2.00e + 02 (3.13e + 00) +	2.00e + 02 (2.64e + 00) +	**1.97e + 02 (7.29e − 02)**
F3	50	7.82e − 02 (4.51e − 02) +	9.01e − 02 (5.63e − 02) +	**1.74e − 02 (1.45e − 02)**
	100	8.92e − 02 (3.41e − 02) ≈	9.03e − 02 (2.44e − 02) +	**5.14e − 02 (1.95e − 02)**
	200	1.00 e − 01 (3.75e − 02) +	1.18e − 01 (5.75e − 02) +	**4.42e − 02 (3.36e − 02)**
F4	50	7.75e − 04 (7.31e − 04) +	7.83e − 04 (8.33e − 04) +	**9.41e − 05 (9.36e − 05)**
	100	8.91e − 04 (4.49e − 04) ≈	9.70e − 04 (5.24e − 04) +	**3.45e − 04 (1.53e − 04)**
	200	1.40e − 03 (9.06e − 04) +	1.64e − 03 (8.71e − 04) +	**7.42e − 04 (2.66e − 04)**
F5	50	2.70e + 00 (3.46e + 00) +	1.95e + 00 (1.97e + 00) +	**9.30e − 02 (1.31e − 01)**
	100	4.38e + 00 (3.68e + 00) +	4.68e + 00 (2.64e + 00) +	**9.51e − 02 (1.88e − 01)**
	200	1.26e + 01 (1.12e + 01) +	9.11e + 00 (7.48e + 00) +	**1.13e − 01 (1.49e − 01)**
F6	50	1.05e + 03 (7.88e + 00) +	1.04e + 03 (5.66e + 01) ≈	**1.03e + 03 (1.67e + 01)**
	100	**1.97e + 03 (4.67e + 01) –**	1.98e + 03 (2.57e + 01) ≈	2.00e + 03 (3.52e + 01)
	200	4.39e + 03 (6.82e + 01) ≈	**4.35e + 03 (9.10e + 01) ≈**	4.37e + 03 (1.06e + 02)
F7	50	9.10 e + 02 (2.55e − 01) ≈	9.10e + 02 (4.71e − 01) ≈	**9.10e + 02 (1.80e − 02)**
	100	9.10e + 02 (1.15e + 00) ≈	9.11e + 02 (8.65e − 01) +	**9.10e + 02 (2.04e − 02)**
	200	9.11e + 02 (1.18e + 00) +	9.11e + 02 (2.76e − 01) +	**9.10e + 02 (9.92e − 03)**
+ / ≈ / –		15 / 5 / 1	16 / 5 / 0	N/A
Average ranking		2.26	2.52	1.21

Wilcoxon rank sum test at a 95% significant level, DDCSAEA significantly outperforms SHPSO, TL-SSLPSO, and SAMSO on at least 22 test instances and significantly outperforms DESO and TS-DDEO on at least 19 test instances. Furthermore, according to the average rankings computed by the Friedman test, the proposed DDCSAEA ranks first among the contestants, indicating its strong computational efficiency and robustness.

To be more specific, as shown in Table 5, DDCSAEA outperforms the other comparative algorithms for the 30-dimensional unimodal problem F1. However, as the dimensionality of the problem increases, DDCSAEA's performance improves and significantly outperforms SHPSO, TL-SSLPSO, SAMSO, and DESO, whereas its computational efficiency necessitates being further improved in contrast to TS-DDEO. We speculate that this is mainly due to the surrogate-assisted dimension-by-dimension crossover strategy in TS-DDEO. Underlying the dimension-by-dimension crossover strategy on the current optimal solution with a portion of the optimal sample set, TS-DDEO updates the global optimum by yielding as many candidate solutions as possible in the neighborhood of the optimal solution. This in turn enhances the algorithm's local exploitation to the neighborhood of the optimum and significantly increases the algorithm's convergence accuracy, especially for unimodal problems. Nonetheless, DDCSAEA can obtain significantly better solutions than TS-DDEO for the multimodal problems F4, F5, and F7.

For F2, whose global optimum locates in a narrow basin, the proposed DDCSAEA yields superior results over SHPSO, TL-SSLPSO, and SAMSO in 30- and 50-dimensional cases, while its final results are not as good as that of DESO and TS-DDEO. However, under the scenario of 100- and 200-dimensional instances, DDCSAEA has achieved the best solutions among all the competitors. This indicates that the neighboring prior knowledge of the optimal solution in the original high-dimensional problem space can be effectively learned through feature reduction, subspace local search and feature reconstruction as the increase of problem complexity.

For F3, which features a symmetric multimodal fitness landscape, DDCSAEA defeats the other five competitors in 50-, 100-, and 200-dimensional cases and obtains significantly better results. From Table 5, one can note that

Table 5 Comparison results of DDCSAEA against five advanced algorithms on the selected problems

No.	D	SHPSO Mean (Std.)	TL-SSLPSO Mean (Std.)	SAMSO Mean (Std.)	DESO Mean (Std.)	TS-DDEO Mean (Std.)	DDCSAEA Mean (Std.)
F1	30	4.07e − 01 (2.35e − 01) +	7.27e − 05 (1.66e − 04) –	5.70e − 03 (7.20e − 03) –	**1.70e − 18 (8.05e − 19) –**	8.32e − 14 (9.01e − 14) –	5.76e − 03 (1.15e − 03)
	50	6.76e + 00 (2.41e + 00) +	5.79e − 01 (1.55e + 00) ≈	5.29e − 01 (2.30e − 01) +	5.15e − 10 (4.45e − 10) –	**7.41e − 11 (5.18e − 11) –**	1.46e − 02 (1.79e − 02)
	100	1.22e + 02 (3.22e + 01) +	2.41e + 01 (1.46e + 01) +	6.62e + 01 (1.96e + 01) +	2.01e + 00 (1.10e + 00) ≈	**6.04e − 06 (5.94e − 06) –**	2.15e − 02 (4.05e − 02)
	200	1.12e + 03 (2.18e + 02) +	6.02e + 02 (1.24e + 02) +	1.48e + 03 (1.58e + 02) +	8.26e + 02 (1.91e + 02) +	6.24e − 01 (3.40e − 01) +	**1.98e − 01 (5.94e − 02)**
F2	30	4.65e + 01 (1.25e + 01) +	2.95e + 01 (1.13e + 01) +	2.83e + 01 (7.05e − 01) –	**2.45e + 01 (1.18e + 00) –**	2.70e + 01 (6.58e − 01) –	2.89e + 01 (1.41e − 01)
	50	1.60e + 02 (3.44e + 01) +	6.81e + 01 (3.06e + 01) ≈	4.99e + 01 (9.90e − 01) +	**4.65e + 01 (6.00e − 01) –**	4.80e + 01 (7.28e − 01) –	4.85e + 01 (6.58e − 03)
	100	8.73e + 02 (2.08e + 02) +	1.30e + 02 (4.98e + 01) +	2.93e + 02 (3.72e + 01) +	1.00e + 02 (1.60e + 00) +	1.05e + 02 (1.05e + 01) +	**9.80e + 01 (1.19e − 02)**
	200	2.72e + 03 (1.50e + 03) +	9.07e + 02 (1.29e + 02) +	1.09e + 03 (1.33e + 02) +	3.43e + 02 (3.28e + 01) +	2.66e + 02 (4.23e + 01) +	**1.97e + 02 (7.29e − 02)**
F3	30	4.00e − 02 (1.00e − 02) +	8.37e − 01 (8.51e − 01) +	5.87e − 01 (3.79e − 01) +	**3.23e − 05 (1.58e − 05) –**	1.54e − 04 (4.33e − 05) –	5.91e − 03 (5.69e − 03)
	50	7.29e − 02 (2.02e − 02) +	1.13e + 00 (7.22e − 01) +	1.53e + 00 (3.12e − 01) +	2.39e − 01 (4.48e − 01) +	1.55e − 01 (2.90e − 01) +	**1.74e − 02 (1.45e − 02)**
	100	1.19e + 00 (1.36e − 01) +	2.25e + 00 (3.68e − 01) +	6.04e + 00 (3.42e − 01) +	2.76e + 00 (3.30e − 01) +	2.51e + 00 (3.19e − 01) +	**5.14e − 02 (1.95e − 02)**
	200	1.59e + 00 (2.06e − 01) +	7.94e + 00 (8.10e − 01) +	1.19e + 01 (3.55e − 01) +	5.82e + 00 (3.60e − 01) +	5.59e + 00 (4.01e − 01) +	**4.42e − 02 (3.36e − 02)**
F4	30	1.80e − 03 (1.30e − 03) +	2.30e − 03 (4.9e − 03) +	4.47e − 01 (1.29e − 01) +	3.07e − 05 (3.10e − 05) +	3.40e − 03 (3.90e − 03) +	**1.25e − 05 (2.43e − 05)**
	50	1.60e − 03 (5.14e − 04) +	3.68e − 02 (5.91e − 02) +	6.33e − 01 (1.03e − 01) +	8.80e − 03 (4.80e − 03) +	3.44e − 02 (1.62e − 02) +	**9.41e − 05 (9.36e − 05)**
	100	6.14e − 03 (2.74e − 03) +	9.07e − 01 (7.89e − 02) +	1.07e + 00 (2.12e − 02) +	1.02e + 00 (9.07e − 02) +	2.51e + 00 (3.19e − 01) +	**3.45e − 04 (1.53e − 04)**
	200	2.13e + 01 (6.00e − 03) +	8.52e + 00 (2.94e − 00) +	8.82e + 00 (1.28e + 00) +	1.56e + 01 (3.06e + 00) +	7.60e − 01 (6.71e − 02) +	**7.42e − 04 (2.66e − 04)**
F5	30	1.86e + 02 (4.40e + 01) +	9.47e + 01 (2.05e + 01) +	2.68e + 01 (9.90e + 00) +	2.21e + 02 (1.51e + 01) +	1.99e + 02 (1.31e + 01) +	**6.22e − 02 (1.19e − 01)**

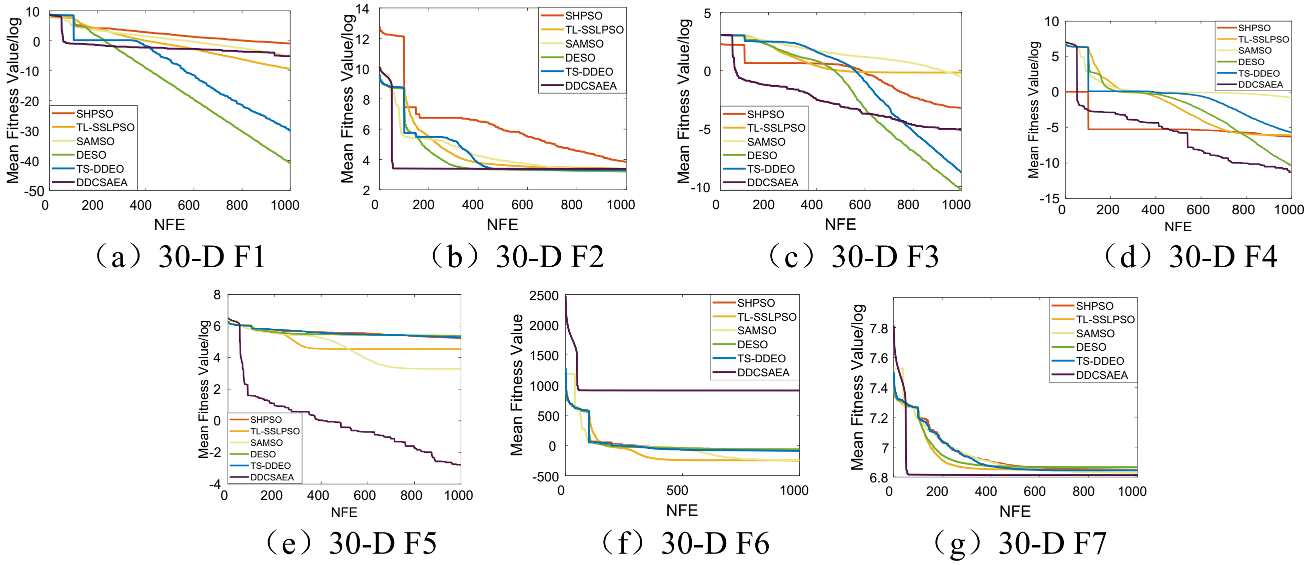
Table 5 (continued)

No.	D	SHPSO Mean (Std.)	TL-SSLPSO Mean (Std.)	SAMSO Mean (Std.)	DESO Mean (Std.)	TS-DDEO Mean (Std.)	DDCSAEA Mean (Std.)
	50	3.85e + 02 (7.27e + 01) +	1.80e + 02 (5.29e + 01) +	3.99e + 01 (1.53e + 01) +	3.97e + 02 (1.58e + 01) +	3.79e + 02 (3.34e + 01) +	**9.30e − 02 (1.31e − 01)**
	100	8.89e + 02 (1.01e + 02) +	3.25e + 02 (1.45e + 02) +	4.35e + 03 (7.95e + 02) +	8.80e + 02 (3.75e + 01) +	8.55e + 02 (4.62e + 01) +	**9.51e − 02 (1.88e − 01)**
	200	1.85e + 03 (1.14e + 02) +	2.08e + 03 (6.48e + 01) +	1.68e + 03 (1.27e + 02) +	1.84e + 03 (3.42e + 01) +	1.82e + 03 (6.29e + 01) +	**1.13e − 01 (1.49e − 01)**
F6	30	− 8.32e + 01 (2.88e + 01)–	**− 2.52e + 02 (2.12e + 01)–**	− 2.39e + 02 (1.93e + 01)–	− 6.36e + 01 (2.09e + 01) –	− 8.92e + 01 (1.66e + 01) –	5.52e + 02 (3.13e + 01)
	50	1.17e + 02 (2.78e + 01) –	**− 1.80e + 02 (3.74e + 01)–**	− 1.51e + 02 (3.15e + 01)–	2.00e + 02 (3.75e + 01) –	1.11e + 02 (2.30e + 01) –	1.03e + 03 (1.67e + 01)
	100	7.94e + 02 (5.74e + 01) –	8.12e + 02 (5.53e + 01)–	7.73e + 02 (5.25e + 02) –	1.23e + 03 (7.19e + 01) –	**7.56e + 02 (8.60e + 01) –**	2.00e + 03 (3.52e + 01)
	200	5.53e + 03 (2.78e + 02) +	5.55e + 03 (1.67e + 02) +	4.95e + 03 (1.67e + 02) +	4.88e + 03 (1.43e + 02) +	5.04e + 03 (1.95e + 02) +	**4.37e + 03 (1.06e + 02)**
F7	30	9.42e + 02 (9.79e + 00) +	9.42e + 02 (1.66e + 01) +	9.22e + 02 (3.50e + 01) +	9.59e + 02 (1.56e + 01) +	9.38e + 02 (1.19e + 01) +	**9.10e + 02 (7.75e − 02)**
	50	9.42e + 02 (9.79e + 00) +	9.96e + 02 (3.16e + 01) +	9.78e + 02 (2.80e + 01) +	1.03e + 03 (2.65e + 01) +	1.00e + 03 (2.12e + 01) +	**9.10e + 02 (1.80e − 02)**
	100	1.43e + 03 (2.94e + 01) +	1.40e + 03 (4.30e + 01) +	1.27e + 03 (3.27e + 01) +	1.34e + 03 (4.79e + 01) +	1.38e + 03 (3.39e + 01) +	**9.10e + 02 (2.04e − 02)**
	200	1.46e + 03 (6.01e + 01) +	1.47e + 03 (1.88e + 01) +	1.31e + 03 (2.94e + 01) +	1.35e + 03 (5.78e + 01) +	1.41e + 03 (3.52e + 01) +	**9.10e + 02 (9.92e − 03)**
+ / ≈ / –		25/0/3	22/2/4	23/0/5	19/1/8	19/0/9	N/A
Average ranking		4.46	3.95	3.86	3.64	3.05	2.04

DDCSAEA is slightly worse than SHPSO, DESO, and TS-DDEO in solving the 30-dimensional instance. This may attribute to DDCSAEA's iterative execution of feature reduction and reconstruction on the current optimal solution and its neighboring candidate solutions for updating the iterative population, resulting in rapid degradation of population diversity. On the contrary, SHPSO, DESO, and TS-DDEO, which all involve global exploration sampling of the original problem space, can effectively maintain the iterative population's diversity as the optimization progresses. Similarly, for the multimodal problems F4 and F5, as shown in Table 5, DDCSAEA holds good robustness as the problem scale varies, and the best solutions obtained are significantly better than those of the other contestants.

For the asymmetric multimodal problems F6-F7, as shown in Table 5, DDCSAEA performs worse than the other five competitors in 30-, 50-, and 100-dimensional cases. However, the performance of DDCSAEA improves significantly when the problem dimension reaches 200, where the best solution obtained by DDCSAEA is significantly better than those of the other five competitors. Particularly, for F7, DDCSAEA has found the best solutions with high qualities against the competitors within 200 dimensions, indicating the high accuracy and good robustness of DDCSAEA in solving such complex problems. Note that DDCSAEA deploys a PCA to reduce the complexity of the target high-dimensional problem space. In this manner, the original problem space can be well smoothed by getting rid of some redundant feature information.

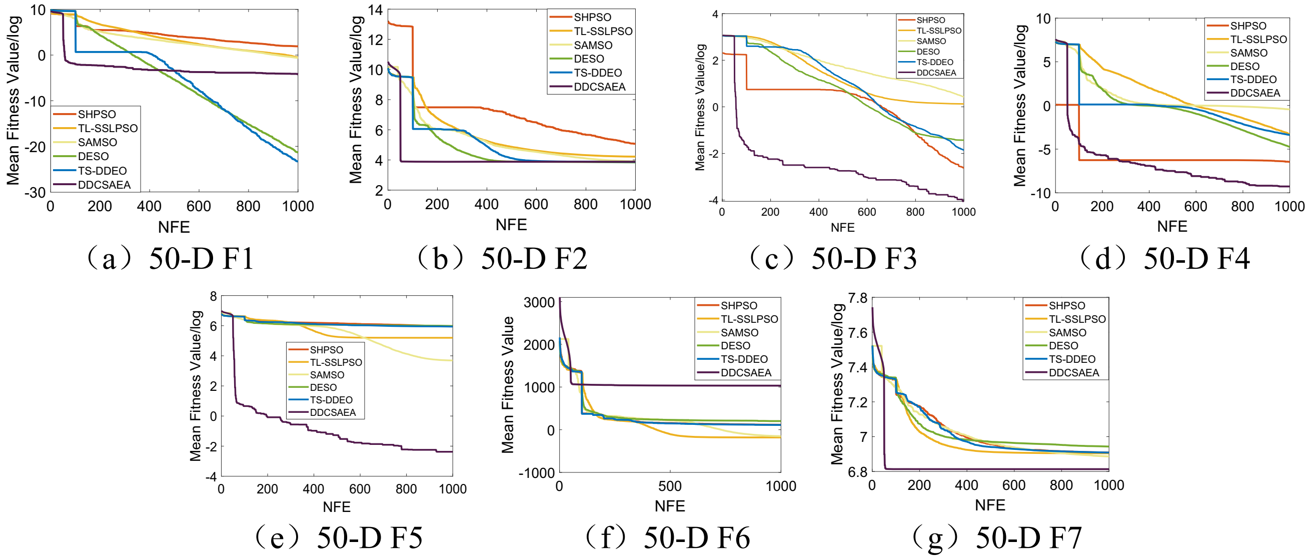
To be more intuitive, Figs. 3, 4, 5 and 6 depict the average performance profiles of each compared algorithm for solving the selected benchmark problems over 20 independent runs. In general, DDCSAEA converges rapidly in the early search



（a）30-D F1 （b）30-D F2 （c）30-D F3 （d）30-D F4

（e）30-D F5 （f）30-D F6 （g）30-D F7

Fig. 3 Convergence profiles of the compared algorithms for 30-D benchmark problems

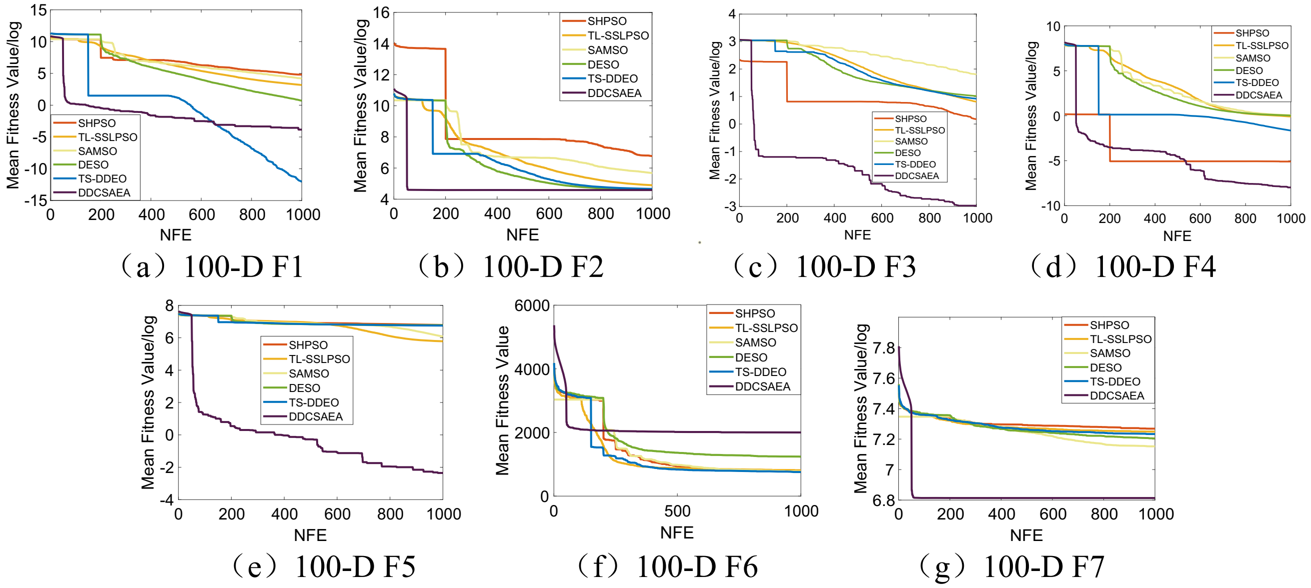


（a）50-D F1 （b）50-D F2 （c）50-D F3 （d）50-D F4

（e）50-D F5 （f）50-D F6 （g）50-D F7

Fig. 4 Convergence profiles of the compared algorithms for 50-D benchmark problems

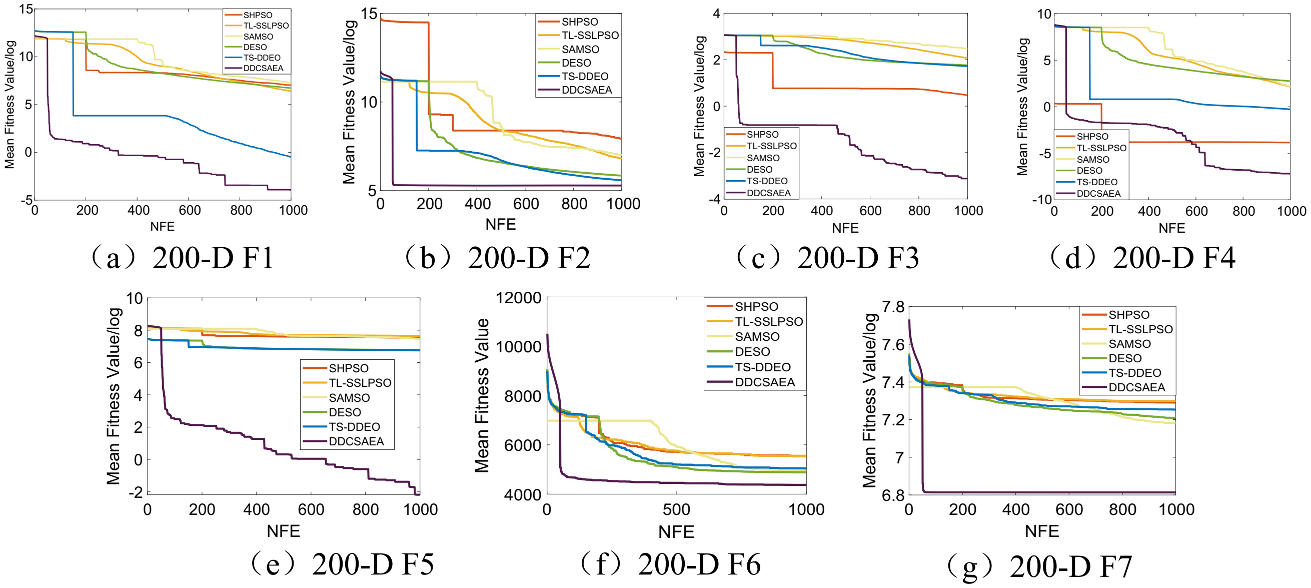
stage and can quickly locate the local optimal solution at the expense of very few real fitness function evaluations, e.g., optimization on F1–F7 of 30 dimensions. Meanwhile, DDCSAEA can still quickly locate the global best solution as the problem scale increases against the other five contestants, such as F2, F3, F4, F5, and F7 with 50 to 200 dimensions. In particular, for F5 and F7, DDCSAEA has better convergence performance than the other five compared algorithms within 1000 real evaluations, and its superiority persists as the problem scale increases. This largely benefits from the compression and simplification of the feature space by the PCA model in DDCSAEA and the surrogate-assisted optimization of the principal feature subspace, which greatly improves the accessibility to prior knowledge of the principal features of the original problem space in the early search stage. Consequently, the promising solutions in the optimal region can be enriched through feature reconstruction. From Fig. 3, we can also find that DDCSAEA is less capable of global exploration in the later search stage than DESO and TS-DDEO, such as F1, F3, F4 and F6. Nevertheless, as shown in Figs. 4, 5 and 6, the convergence performance of DDCSAEA achieves great improvement and is significantly better than the other competitors in F3 and F4 in 50- to 200-dimensional instances. What's more, from Fig. 6, we can observe that DDCSAEA exhibits remarkable convergence performance versus the other five compared algorithms.



（a）100-D F1 （b）100-D F2 （c）100-D F3 （d）100-D F4

（e）100-D F5 （f）100-D F6 （g）100-D F7

Fig. 5 Convergence profiles of the compared algorithms for 100-D benchmark problems



（a）200-D F1 （b）200-D F2 （c）200-D F3 （d）200-D F4

（e）200-D F5 （f）200-D F6 （g）200-D F7

Fig. 6 Convergence profiles of the compared algorithms for 200-D benchmark problems

According to the aforementioned comparative results shown in Table 5 and Figs. 3, 4, 5 and 6, it is possible to conclude that the proposed DDCSAEA holds a fast optimization speed against SHPSO, TL-SSLPSO, SAMSO, DESO, and TS-DDEO under the computation budget of 1000 real fitness function evaluations. In particular, for high-dimensional complex problem solution spaces of 100 to 200 dimensions, DDCSAEA offers significant superiority in terms of convergence performance and robustness.

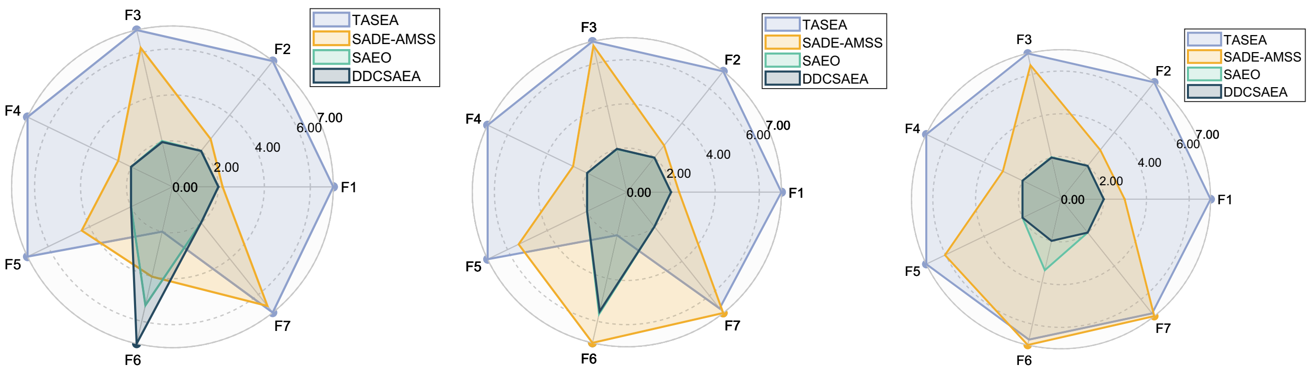
Comparison results of DDCSAEA versus three advanced algorithms with feature reduction

The proposed DDCSAEA is further examined for its computational effectiveness alongside the TASEA, SADE-AMSS and SAEO. These three contestants take advantage of dimension reduction techniques to eliminate redundant or irrelevant features of the target high-dimensional or large-scale feature space, enabling them to deeply mine the prior knowledge following the direction of the most important features within the solution space, thereby improving surrogate

Table 6 Comparison results of DDCSAEA against SAEO algorithm

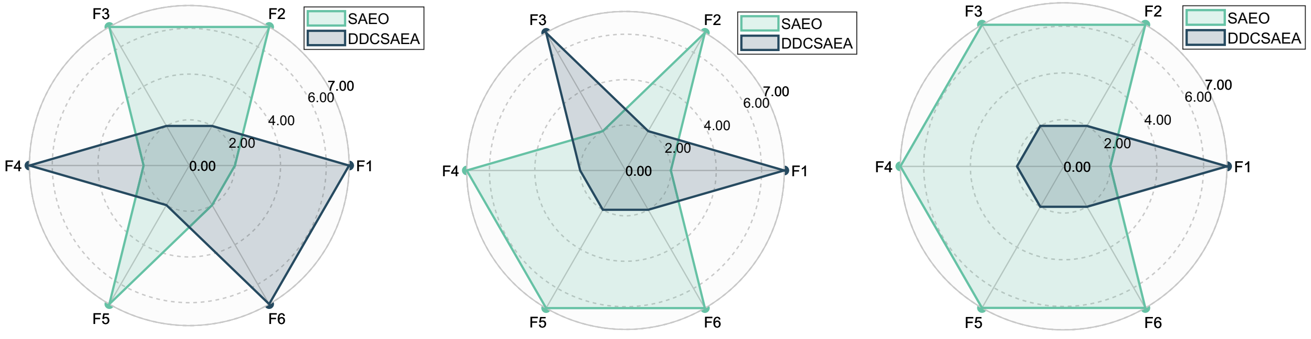
No	D	TASEA Mean (Std.)	SADE-AMSS Mean (Std.)	SAEO Mean (Std.)	DDCSAEA Mean (Std.)
F1	50	1.37e + 03 (2.39e + 02) +	5.04e + 01 (1.83e + 01) +	**8.33e − 15 (3.73e − 14) –**	1.46e − 02 (1.79e − 02)
	100	7.90e + 03 (9.11e + 02) +	5.82e + 02 (1.21e + 02) +	**4.48e − 03 (1.62e − 02) –**	2.15e − 02 (4.05e − 02)
	200	3.92e + 04 (3.92e + 03) +	7.67e + 03 (1.44e + 03) +	4.21e − 02 (2.78e − 02) +	**1.98e − 01 (5.94e − 02)**
F2	50	1.52e + 03 (2.94e + 02) +	2.44e + 02 (9.63e + 01) +	4.90e + 01 (2.54e − 02) +	**4.85e + 01 (6.58e − 03)**
	100	4.85e + 03 (7.00e + 02) +	7.70e + 02 (2.01e + 02) +	9.90e + 01 (6.65e − 02) +	**9.80e + 01 (1.19e − 02)**
	200	1.32e + 04 (1.31e + 03) +	2.66e + 03 (5.68e + 02) +	1.99e + 02 (5.31e − 02) +	**1.97e + 02 (7.29e − 02)**
F3	50	1.70e + 01 (4.21e − 01) +	1.43e + 01 (3.07e + 00) +	1.73e − 01 (4.33e − 01) ≈	**1.74e − 02 (1.45e − 02)**
	100	1.80e + 01 (3.27e − 01) +	1.73e + 01 (2.26e + 00) +	**1.85e − 04 (5.50e − 04) –**	5.14e − 02 (1.95e − 02)
	200	1.86e + 01 (2.88e − 01) +	1.65e + 01 (9.09e − 01) +	1.16e − 01 (9.37e − 02) +	**4.42e − 02 (3.36e − 02)**
F4	50	2.05e + 02 (2.76e + 01) +	2.54e + 01 (6.88e + 00) +	**4.55e − 11 (1.12e − 10) –**	9.41e − 05 (9.36e − 05)
	100	6.06e + 02 (7.02e + 01) +	8.52e + 01 (3.01e + 01) +	1.13e − 02 (2.49e − 02) ≈	**3.45e − 04 (1.53e − 04)**
	200	1.44e + 03 (1.35e + 02) +	2.94e + 02 (4.45e + 01) +	1.19e − 01 (2.88e − 02) +	**7.42e − 04 (2.66e − 04)**
F5	50	4.95e + 02 (2.56e + 01) +	2.36e + 02 (3.71e + 01) +	1.26e − 01 (1.84e − 01) ≈	**9.30e − 02 (1.31e − 01)**
	100	1.09e + 03 (3.23e + 01) +	7.50e + 02 (6.14e + 01) +	1.17e − 01 (2.85e − 01) ≈	**9.51e − 02 (1.88e − 01)**
	200	2.37e + 03 (5.60e + 01) +	1.91e + 03 (7.68e + 01) +	4.00e − 01 (1.98e − 01) +	**1.13e − 01 (1.49e − 01)**
F6	50	**4.08e + 02 (4.36e + 01) –**	6.58e + 02 (1.74e + 02) –	8.17e + 02 (8.92e + 01) –	1.03e + 03 (1.67e + 01)
	100	**1.71e + 03 (1.17e + 02) –**	2.12e + 03 (2.42e + 02) +	2.01e + 03 (1.16e + 02) ≈	2.00e + 03 (3.52e + 01)
	200	6.02e + 03 (1.98e + 02) +	6.11e + 03 (2.78e + 02) +	4.86e + 03 (3.38e + 02) +	**4.37e + 03 (1.06e + 02)**
F7	50	1.18e + 03 (2.65e + 01) +	1.16e + 03 (6.43e + 01) +	**9.10e + 02 (0.00e − 00) –**	9.10e + 02 (1.80e − 02)
	100	1.45e + 03 (2.26e + 01) +	1.48e + 03 (4.51e + 01) +	**9.10e + 02 (2.61e − 14) –**	9.10e + 02 (2.04e − 02)
	200	1.51e + 03 (2.02e + 01) +	1.53e + 03 (3.57e + 01) +	**9.10e + 02 (4.23e − 03) ≈**	9.10e + 02 (9.92e − 03)
+ /≈/−		19/0/2	20/0/1	8/6/7	N/A
Average ranking		3.57	3.14	1.79	1.50

model performance and interpretability. TASEA employs Sammon mapping to project the high-dimensional space into a lower-dimensional subspace, constructing a global GP model and a local GP model for different mutation operators in screening candidate solutions. SADE-AMSS constructs a series of subspaces using PCA and a random feature selection strategy, adaptively switching three subspace search strategies and surrogate modeling according to different optimization stages. SAEO uses an autoencoder for feature reduction and feature reconstruction of the iterative population on the high-dimensional solution space, with a surrogate-assisted greedy sampling in the target high-dimensional space, achieving excellent optimization performance. Table 6 presents the statistical results obtained by TASEA, SADE-AMSS, SAEO, and DDCSAEA on the selected benchmark problems of 50–200 dimensions, with the best results on each instance being highlighted. From Table 6, following the results derived from the pairwise Wilcoxon rank sum test at a 95% significance level, DDCSAEA significantly outperforms TASEA and SADE-AMSS on at least 19 test instances and significantly outperforms SAEO on 8 test instances. According to the Friedman test, DDCSAEA ranks first and attains the best average ranking of 1.50 among all competitors, followed by SAEO, SADE-AMSS and TASEA, indicating the superior performance of DDCSAEA in this category of SAEA with feature reduction. Additionally, for a more visual representation of the comparative performance, the radar charts of DDCSAEA versus TASEA, SADE-AMSS and SAEO are shown in Fig. 7. For the 50-dimensional asymmetric multimodal problem F6, DDCSAEA demonstrates poorer performance compared to TASEA, SADE-AMSS, and SAEO. However, the performance of DDCSAEA exhibits significant improvement as the problem dimension increases to 100-dimensional and 200-dimensional, reaching its optimal performance at 200-dimensional instances. To provide a clearer illustration of the performance differences between DDCSAEA and SAEO, Fig. 8 presents the separate radar charts for these two contestants. From Fig. 8, one can observe that DDCSAEA exhibits significant superiority against SAEO on the selected benchmark problems in terms of comprehensive performance and maintains strong robustness with the problem scale. Here in Fig. 8, the F7 is not marked in the radar charts due to the similar performance of DDCSAEA and SAEO.



(a) 50-D (b) 100-D (c) 200-D

Fig. 7 Radar charts of DDCSAEA versus TASEA, SADE-AMSS and SAEO



(a) 50-D (b) 100-D (c) 200-D

Fig. 8 Radar charts of DDCSAEA versus SAEO

Conclusion

This paper proposes a dual-drive collaboration SAEA namely DDCSAEA for complex high-dimensional expensive optimization problems. In the framework of DDCSAEA, two unsupervised feature learning techniques, PCA and autoencoder, are assembled in tandem to drive its exploration and exploitation across two feature spaces through iterative feature reduction and feature reconstruction. At each iteration of DDCSAEA, a PCA model is trained to reduce the original problem space for getting a principal feature subspace. An RBF-assisted SLPSO is then employed to exploit the optimum of the RBF model. After that, the neighboring samples to the RBF optimum in the feature subspace further undergo differential mutation and crossover to explore some promising candidate solutions for enriching the structural prior information of the optimal region. An autoencoder is thereafter trained to perform feature reconstruction on these candidates to project them into the target high-dimensional space. Finally, these candidate samples and current optimal samples after the feature reconstruction are prescreened based on the nearest neighbor principle to determine the infill samples for real evaluation. DDCSAEA is evaluated on a widely used test suit with various fitness landscapes at different dimensional scales. Experimental results demonstrate that DDCSAEA features superior convergence performance and better robustness over eight state-of-the-art algorithms under the scenario of 30-, 50-, 100-, and 200- dimensional instances.

However, we also note that DDCSAEA performs poorly in dealing with some of the medium and low-dimensional test problems. We speculate that the fixed feature subspace scale leads to the distortion of the feature subspace optimization, resulting in the wrong convergence. In our future work, the adaptive mechanism of the feature reduction scale to the original problem scale will be considered in conjunction with fitness landscape analysis to further improve the applicability of the DDCSAEA. Applying DDCSAEA to the solution of real-world expensive problems is also included in our future work.

Acknowledgements We would like to appreciate Handing Wang, Meiji Cui, and Zan Yang for sharing the MATLAB Codes of SADE-AMSS, SAEO, and TASEA. This work was supported in part by the National Natural Science Foundation of China (Grant Number 62106237), the Joint Funds of the National Natural Science Foundation of China (Grant Number U21A20524), the Shanxi Province Science Foundation for Youths (Grant Number 202203021222057), and the Natural Science Research Project of Shanxi Province (Grant Number 202103021224218).

Funding This work was not supported by this funding.

Data availability The data are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest that could have appeared to influence the work reported in this paper.

References

1. Baysal O, Eleshaky ME (1992) Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. AIAA J 30(3):718–725
2. Park S, Na J, Kim M, Lee JM (2018) Multi-objective bayesian optimization of chemical reactor design using computational fluid dynamics. Comput Chem Eng 119:25–37
3. Elsken T, Metzen JH, Hutter F (2019) Neural architecture search: a survey. J Mach Learn Res 20(1):1997–2017
4. Liu Y, Sun Y, Xue B, Zhang M, Yen GG, Tan KC (2023) A survey on evolutionary neural architecture search. IEEE Trans Neural Netw Learn Syst 34(2):550–570
5. Zhao W, Gupta A, Regan CD, Miglani J, Kapania RK, Seiler PJ (2019) Component data assisted finite element model updating of composite flying-wing aircraft using multi-level optimization. Aerosp Sci Technol 95:105486
6. Manca AG, Pappalardo CM (2020) Topology optimization procedure of aircraft mechanical components based on computer-aided design, multibody dynamics, and finite element analysis[C]//Advances in Design, Simulation and Manufacturing III: Proceedings of the 3rd International Conference on Design, Simulation, Manufacturing: The Innovation Exchange, DSMIE-2020, June 9–12, 2020, Kharkiv, Ukraine–Volume 2: Mechanical and Chemical Engineering. Springer International Publishing, pp 159–168
7. Liu H, Ong Y-S, Cai J (2017) A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. Struct Multidisc Optim 57(1):393–416
8. Wang X, Wang G, Song B, Wang P, Wang Y (2019) A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems. IEEE Trans Evol Comput 23(5):815–827
9. Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. Inf Sci 454–455:59–72
10. Tong H, Huang C, Minku LL, Yao X (2021) Surrogate models in evolutionary single-objective optimization: a new taxonomy and experimental study. Inf Sci 562:414–437
11. Alexander F, Andy K (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45:50–79
12. 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
13. Buche D, Schraudolph N, Koumoutsakos P (2005) Accelerating evolutionary algorithms with Gaussian process fitness function models. IEEE Trans Syst Man Cybern Part C 35(2):183–194
14. Tian J, Tan Y, Zeng J, Sun C, Jin Y (2019) Multiobjective infill criterion driven Gaussian process-assisted particle swarm optimization of high-dimensional expensive problems. IEEE Trans Evol Comput 23(3):459–472
15. I. Loshchilov, M. Schoenauer, M.Sebag, Comparison-Based Optimizers Need Comparison-Based Surrogates, Parallel Problem Solving from Nature–PPSN XI. 6238 (2010) 364–373.
16. Poloczek J, Kramer O (2013) Local SVM constraint surrogate model for self-adaptive evolution strategies, vol 8077. Springer, Berlin, pp 164–175
17. Krithikaa M, Mallipeddi R (2016) Differential evolution with an ensemble of low-quality surrogates for expensive optimization problems. IEEE Congress Evol Comput (CEC) 2016:78–85
18. Wang L, Yao Y, Wang K, Adenutsi CD, Zhao G, Lai F (2021) A novel surrogate-assisted multi-objective optimization method for well control parameters based on tri-training. Nat Resour Res 30:4825–4841
19. Tong H, Huang C, Liu J, Yao X (2019) Voronoi-based efficient surrogate-assisted evolutionary algorithm for very expensive problems. IEEE Congress Evol Comput (CEC) 2019:1996–2003
20. Guo D, Jin Y, Ding J, Chai T (2019) Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. IEEE Trans Cybern 49(3):1012–1025
21. Yu M, Liang J, Wu Z, Yang Z (2021) A twofold infill criterion-driven heterogeneous ensemble surrogate-assisted evolutionary algorithm for computationally expensive problems. Knowl Based Syst 236:107747
22. Yang Z, Qiu H, Gao L, Jiang C, Zhang J (2019) Two-layer adaptive surrogate-assisted evolutionary algorithm for high-dimensional computationally expensive problems. J Global Optim 74(2):327–359
23. 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
24. 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 Petrol Sci Eng 192:0920–4105
25. Gu H, Wang H, Jin Y (2022) Surrogate-assisted differential evolution with adaptive multi-subspace search for large-scale expensive optimization. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2022.3226837
26. Feng L, Shang Q, Hou Y, Tan KC, Ong Y-S (2023) Multispace evolutionary search for large-scale optimization with applications to recommender systems. IEEE Trans Artif Intell 4(1):107–120
27. Li X, Li S (2021) An adaptive surrogate-assisted particle swarm optimization for expensive problems. Soft Comput 25:15051–15065
28. Cui M, Li L, Zhou M, Abusorrah A (2022) Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems. IEEE Trans Evol Comput 26(4):676–689

29. 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
30. Younis A, Dong Z (2010) Trends, features, and tests of common and recently introduced global optimization methods. Eng Optim 42(8):691–718
31. Díaz-Manríquez A, Pulido GT, Coello CA (2017) Comparison of metamodeling techniques in evolutionary algorithms. Soft Comput 21:5647–5663
32. Yu H, Tan Y, Sun C, Zeng J (2019) A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization[J]. Knowl-Based Syst 163:14–25
33. Gutmann HM (2001) A radial basis function method for global optimization. J Global Optim 19(3):201–227
34. Bookstein FL (1989) Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans Pattern Anal Mach Intell 11(6):567–585
35. Regis RG (2014) Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng Optim 46:218–243
36. Lever J, Krzywinski M, Altman N (2017) Principal component analysis. Nat Methods 14:641–642
37. Bengio Y, Courville AC, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35:1798–1828
38. Al-Hmouz R, Pedrycz W, Balamash A, Morfeq A (2022) Logic-oriented autoencoders and granular logic autoencoders: developing interpretable data representation. IEEE Trans Fuzzy Syst 30(3):869–877
39. Bengio Y (2007) Learning deep architectures for AI. Found Trends Mach Learn 2:1–127
40. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
41. Bentley PJ, Lim SL, Gaier A, Tran L (2022) Evolving through the looking glass: learning improved search spaces with variational autoencoders[C]. In: Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I. Cham: Springer International Publishing, pp 371–384
42. Jeffrey H, Goodfellow I, Bengio Y, Courville A (2018) Deep learning. Genet Program Evolvable Mach 19:305–307
43. Hayat M, Bennamoun M, An S (2015) Deep reconstruction models for image set classification. IEEE Trans Pattern Anal Mach Intell 37(4):713–727
44. Helton JC, Davis FJ (2003) Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliab Eng Syst Saf 81(1):23–69
45. Wang Z, Zhang Q, Ong Y-S, Yao S, Liu H, Luo J (2023) Choose appropriate subproblems for collaborative modeling in expensive multiobjective optimization. IEEE Trans Cybern 53(1):483–496
46. Yu H, Kang L, Tan Y, Sun C, Zeng J (2020) Truncation-learning-driven surrogate assisted social learning particle swarm optimization for computationally expensive problem. Appl Soft Comput 97(Part A):106812
47. Huixiang Z, Wenyin G, Ling W (2021) Data-driven evolutionary sampling optimization for expensive problems. J Syst Eng Electron 32(2):318–330
48. Zhen H, Gong W, Wang L, Ming F, Liao Z (2023) Two-Stage data-driven evolutionary optimization for high-dimensional expensive problems. IEEE Trans Cybern 53(4):2368–2379
49. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[J]. KanGAL report 2005005:2005