Dimensionality Reduction Evolutionary Framework for Solving High-Dimensional Expensive Problems

SONGWei[1], ZOUFucai[2]
Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, WuXi, China[1]
School of Artificial Intelligence and Computer Science, Jiangnan University, WuXi, China[2]

***Abstract*—Most of improvement strategies for surrogate-assisted optimization algorithms fail to help the population quickly locate satisfactory solutions. To address this challenge, a novel framework called dimensionality reduction surrogate-assisted evolutionary (DRSAE) framework is proposed. DRSAE introduces an efficient dimensionality reduction network to create a low-dimensional search space, allowing some individuals to search in the population within the reduced space. This strategy significantly lowers the complexity of the search space and makes it easier to locate promising regions. Meanwhile, a hierarchical search is conducted in the high-dimensional space. Lower-level particles indiscriminately learn from higher-level peers, correspondingly the highest-level particles undergo self-mutation. A comprehensive comparison between DRSAE and mainstream HEPs algorithms was conducted using seven widely used benchmark functions. Comparison experiments on problems with dimensionality increasing from 50 to 200 further substantiate the good scalability of the developed optimizer.**

***Keywords*—Dimensionality reduction; high-dimensional expensive optimization; Surrogate-assisted model**

I. Introduction

Evolutionary algorithms (EAs) have been effectively utilized in addressing optimization problems owing to their simplicity and efficiency. In the era of big data, an increasing number of optimization problems involve a substantial quantity of decision variables, such as traffic vehicle scheduling [1], routing network issues [2], and biological gene identification [3]. When confronted with high-dimensional optimization problems comprising hundreds or even thousands of decision variables, traditional EAs often struggle to identify the optimal solution within the limited number of fitness evaluation (FE) iterations [4]. In high-dimensional expensive problems (HEPs), not only does the search space undergo exponential expansion and increased complexity, but also the time required for evaluating the objective function becomes exceedingly costly. Therefore, simply applying existing EAs to solve HEPs is both time-consuming and inefficient.

Incorporating dimensionality reduction techniques into high-dimensional optimization problems, thereby reducing the complex high-dimensional search space to a lower-dimensional space with higher information density, represents an effective approach for addressing high-dimensional challenges. In the reduced dimensional space, it becomes easier to swiftly identify promising regions, facilitating the generation of high-quality offspring at an accelerated pace. The application of Sammon mapping [5] for dimensionality reduction in EAs problem is motivated by its ability to preserve adjacent structures. However, literature [6] suggests that the performance of Sammon mapping often falters when dealing with intricate datasets. To tackle this issue, SAEO [7] proposed an evolutionary algorithm based on autoencoders (AE), which incorporates a reconstruction stage capable of restoring low-dimensional particles to their original high-dimensional space for real fitness evaluation. Nevertheless, AE necessitates a substantial amount of historical data before reaching training maturity and initial historical data is frequently comprised of poorly performing particles, potentially leading to biased evolutionary directions towards unpromising regions. Furthermore, training the neural network using backpropagation demands a significant time investment.

In response to the challenge of identifying promising regions in HEPs amidst the "dimension disaster," there is a pressing demand for an effective dimensionality reduction technique that can efficiently minimize the search space, facilitating the rapid identification of promising regions while retaining the capacity to reconstruct meaningful information in the original space. This paper presents a Dimensionality Reduction Surrogate-Assisted Evolutionary (DRSAE) framework. Its specific contributions are as follows: 1) It initiates the exploration of low-dimensional space, enhancing precise dimensionality reduction and reconstruction of spatial particles by employing an Extreme Learning Machine based on autoencoder (ELM-AE) for space particles. It uncovers implicit information in the low-dimensional space through evolutionary search variations within that space. 2) The hierarchical search for particles in high-dimensional space aims to balance convergence and diversity, with lower-level particles focusing more on exploration and higher-level particles emphasizing development. Finally, the algorithm's performance is further validated through experiments.

II. Related Work

A. High-Dimensional Expensive Problems (HEPs)

The design of contemporary complex products often entails addressing a multitude of high-dimensional and costly optimization challenges, necessitating thousands of precise simulation analyses that consume substantial computing resources [8]. In this study, we focus on a category of minimization problems:

$$minimize: f(\boldsymbol{x}) \tag{1}$$

The National Natural Science Foundation of China (62076110)
The National Foreign Experts Program (DL2023144001L)
The Natural Science Foundation of Jiangsu Province (BK20181341)
The Fundamental Research Funds for the Central Universities (JUSRP221027)

$$subject\ to: \boldsymbol{x} \le \boldsymbol{x} \le \boldsymbol{x} \tag{2}$$

Above the (1) and (2), $\boldsymbol{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$ represents a d dimensional decision vector within the feasible search space $\mathbb{R}^d$, and $f(\cdot)$ denotes the objective function used for fitness evaluation. Additionally, $\boldsymbol{x}$ and $\boldsymbol{x}$ correspond to the lower and upper bound vectors of the search space, respectively. In cases where the value of d is exceedingly large and evaluating $f(\cdot)$ requires significant time and resources, these issues are classified as HEPs.

Whether dealing with a continuous or combinatorial single- or multi-objective problem, agent-assisted evolutionary algorithms are widely recognized as a promising approach for addressing HEPs. The essence of these algorithms lies in developing suitable agent models based on historical data samples to approximate the true objective function. In terms of computational resource requirements, the evaluation cost of these agent models is significantly lower than that of the real model, enabling them to effectively pre-screen candidate solutions. Based on the predictions generated by the agent model, certain candidate individuals are chosen for re-evaluation using the real model.

B. Extreme Learning Machine-Autoencoder (ELM-AE)

ELM-AE is a single-hidden-layer neural network model, in which the number of nodes in the input and output layers are identical. The input and output of the model represent positions within a high-dimensional space, while the hidden layer's output represents positions within a lower-dimensional space following dimensionality reduction; hence, the dimension or number of nodes in the hidden layer is reduced.

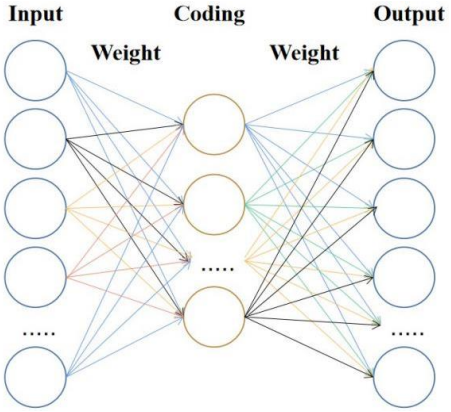

Fig. 1. ELM-AE.

As depicted in Fig. 1, the input weights are initially initialized with random values, employing a unitary orthogonal matrix chosen at random as the input weight [9]. This selection serves to maintain the Euclidean distance between data points and exhibits favorable generalization properties.

The hidden layer output $\boldsymbol{H} = G(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{b})$ is obtained by applying the activation function $G(\cdot)$ to the result of multiplying sample $\boldsymbol{X}$ by $\boldsymbol{A}$, adding the bias $\boldsymbol{b}$, and passing it through. For a given dataset $\{(\boldsymbol{x}_i, \boldsymbol{t}_i)\}_{i=1}^{N}$, $\boldsymbol{H}$ denotes the output of the hidden layer:

$$\boldsymbol{H} = \begin{bmatrix} g_1(\boldsymbol{x}_1\boldsymbol{a}_1^T + b_1) & \cdots & g_L(\boldsymbol{x}_1\boldsymbol{a}_m^T + b_1) \\ \vdots & \ddots & \vdots \\ g_1(\boldsymbol{x}_N\boldsymbol{a}_1^T + b_N) & \cdots & g_L(\boldsymbol{x}_N\boldsymbol{a}_m^T + b_N) \end{bmatrix} \tag{3}$$

In (3), N represents the number of training samples, L denotes the number of nodes in the hidden layer, $g_1, g_2 ... g_L$ are indicative of the activation functions associated with the hidden layer nodes, and $\boldsymbol{x}_i$ signifies the i-th training data. Essentially, $\boldsymbol{H}$ serves as a feature mapping representation of the training data $\boldsymbol{X}$ following matrix operations and activation functions. The aforementioned process can be succinctly represented in matrix form as follows:

$$\boldsymbol{H} = \boldsymbol{G(XA + b)} \tag{4}$$

$\boldsymbol{T}$ represents the target matrix corresponding to the training samples:

$$\boldsymbol{T} = \begin{bmatrix} t_{11} & \cdots & t_{1M} \\ \vdots & \ddots & \vdots \\ t_{N1} & \cdots & t_{NM} \end{bmatrix} \tag{5}$$

M represents the number of nodes in the output layer, while $\boldsymbol{t}_i$ denotes the corresponding target value for $\boldsymbol{x}_i$. The primary objective of ELM is to minimize the error associated with fitting the expired output $\boldsymbol{T}$, as per (6):

$$\boldsymbol{H\beta} = \boldsymbol{T} \tag{6}$$

The analytical solution for the weight matrix connecting the hidden layer and the output layer is derived [10]:

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{7}$$

The Moore-Penrose generalized inverse matrix $\boldsymbol{H}^{\dagger}$ serves as an effective tool for rapidly adjusting $\boldsymbol{X}$ to $\boldsymbol{T}$. As the characteristic representation of $\boldsymbol{X}$, $\boldsymbol{H}$ can analytically compute the output weight $\boldsymbol{\beta}$ that maps to $\boldsymbol{T}$.

$$\boldsymbol{H\beta} = \boldsymbol{X} \tag{8}$$

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{X} \tag{9}$$

Importantly, when the ELM-AE target value equals $\boldsymbol{X}$ itself, the computed $\boldsymbol{H}$ serves as a high-quality representation of $\boldsymbol{X}$ and can be reconstructed to $\boldsymbol{X}$ using the calculated $\boldsymbol{\beta}$ in (9).

III. Dimensionality Reduction Surrogate-Assisted Evolutionary Framework

This paper proposes dimensionality reduction techniques to HEPs, enabling the algorithm to create a low-dimensional space for extracting implicit information for local search. Simultaneously, it preserves the multi-population optimization strategies within the high-dimensional space. Furthermore, it is viable to substitute the costly evaluation function in the high-dimensional space with a surrogate model, such as a radial basis function (RBF) network [11], to capture the global contour of the fitness landscape and facilitate rapid identification of the region containing the global optimum by the population.

Subsequently, we present an overview of our algorithm's framework, which integrates our enhanced ELM-AE dimensionality reduction network and hierarchical particle search strategy within high-dimensional space. We then provide a detailed exposition of two parallel search paths before concluding with an analysis of the algorithm's time complexity.

A. DRSAE Overall Framework

Fig. 2 illustrates the comprehensive process of DRSAE. Prior to activating the RBF agent model, this study employs the classical differential evolution algorithm [12] to iteratively refine the initial population, thereby gathering sufficient real evaluation data pairs conducive to training the initial agent model in a more favorable region. Upon activation of the agent model, the population is sorted based on fitness value and divided into two subpopulations. The subpopulation with lower fitness values undergoes hierarchical search in the original high-dimensional space, while the other subpopulation conducts a search in low-dimensional space through ELM-AE. The size of each subpopulation will dynamically adjust as follows:

In (10), Z denotes the population size, Z_1 denotes the size of the high-dimensional subpopulation, Z_2 denotes the size of the low-dimensional subpopulation, FE_{max} represents the maximum number of fitness evaluations, and FE_{cur} represents the current number of consumed fitness evaluations.

B. Revised ELM-AE-Guided Exploration of Low-Dimensional Spaces

It exploits the advantages of rapid training in ELM and data representation learning in AE. This investigation utilizes ELM-AE to tackle HEPs and introduces enhancements. As illustrated in Fig. 1, the input layer and output layer represent the positions of high-dimensional space particles in the original space, while the hidden layer signifies the positions of low-dimensional space particles. The input weight matrix $\boldsymbol{A}$ is typically generated randomly. In this study, $\boldsymbol{A}$ is a unit random orthogonal matrix, which can be denoted as:

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I} \tag{10}$$

Where $\boldsymbol{I}$ represents the identity matrix.

$$|\boldsymbol{A}_{ij}| \le 1 \tag{11}$$

$$\begin{aligned} Z_1 &= min(Z, Z \times (\frac{FE_{max} - FE_{cur}}{FE_{\max}})^{200/d}) \\ Z_2 &= Z - Z_1 \end{aligned} \tag{12}$$

In (12), $|A_{ij}|$ denotes the element in the i-th row and j-th column of matrix $\boldsymbol{A}$, while $|\cdot|$ represents the absolute value function. Each element of matrix $\boldsymbol{A}$ possesses an absolute value less than 1, and the columns are mutually orthogonal. The unit random orthogonal matrix A can effectively preserve the Euclidean distances between data points [9] and demonstrate enhanced generalization. As per (4), N particles in $\boldsymbol{X}$ undergo dimensional reduction to a low-dimensional space denoted by $\boldsymbol{H}$ for brevity as follows:

$$\begin{aligned} \boldsymbol{b} &= \boldsymbol{0} \\ g(\boldsymbol{X}) &= \boldsymbol{X} \end{aligned} \tag{13}$$

The bias vector $\boldsymbol{b}$ is a zero vector, and the activation function $g(\cdot)$ is a linear function. The output weight matrix beta is determined from the low-dimensional space to the high-dimensional space using (9), which ensures an effective transformation between different dimensional spaces.

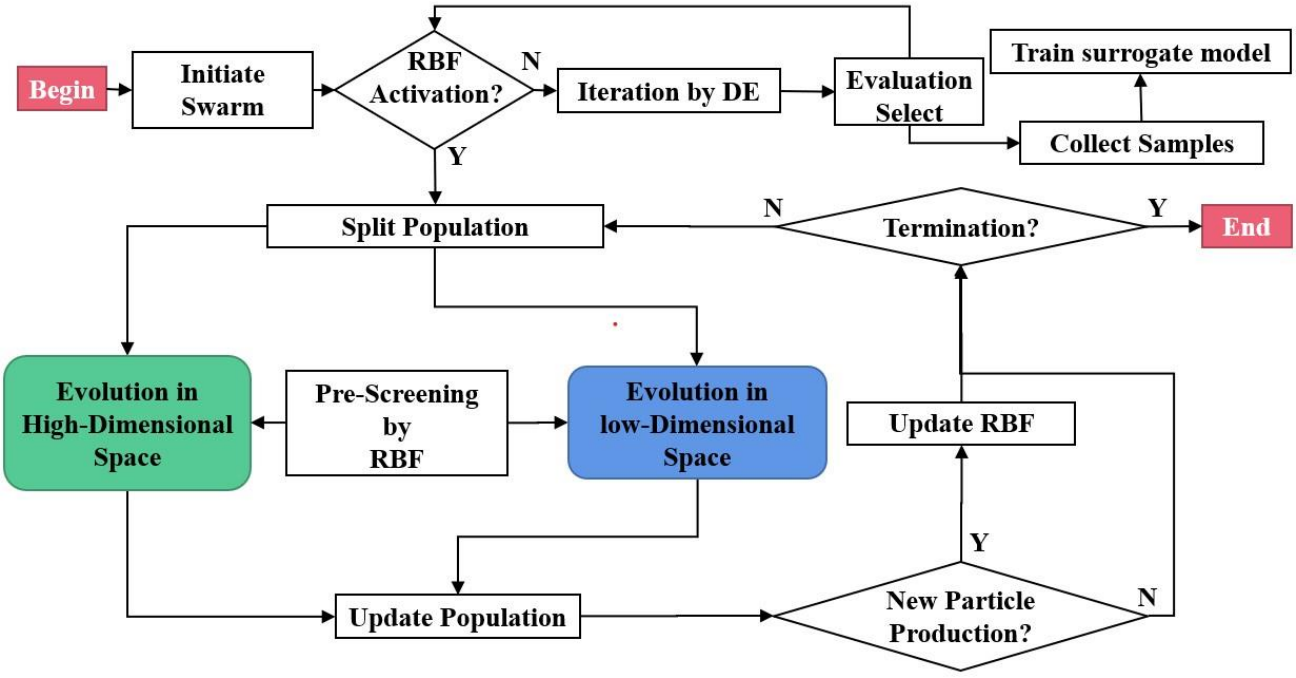

Fig. 2. DRSAE process schematic.

In numerous practical scenarios, it is crucial to acknowledge that $\boldsymbol{H}$ may not be a square matrix ($L \neq N$), leading to the computed $\boldsymbol{H}^{\dagger}$ being the generalized inverse matrix of $\boldsymbol{H}$ rather than the true inverse matrix. As a result, there is a loss of high-dimensional information. This investigation enhances the architecture of ELM-AE to ensure that $\boldsymbol{H}$'s output forms a square matrix. Specifically, setting the number of input particles equal to the dimension of the low-dimensional space enables obtaining the true $\boldsymbol{H}^{\dagger}$, thereby circumventing any information loss upon restoration of the high-dimensional space.

It is apparent that the reduction in variation within low-dimensional space leads to a corresponding decrease in error during projection into high-dimensional space. In order to address the potential dominance of reduction errors over evolutionary processes, a Gaussian field variation algorithm as described in [12] has been incorporated within the low-dimensional space to enhance local fine-tuning search capabilities. The subsequent section will present the updated formula for low-dimensional particles.

$$\begin{aligned} \overrightarrow{\boldsymbol{H}_i} &= \overrightarrow{\boldsymbol{H}_i} + \vec{\boldsymbol{\Delta}} \\ \vec{\boldsymbol{\Delta}} &= [\Delta_1, \Delta_2, ..., \Delta_L] \end{aligned} \tag{14}$$

$\overrightarrow{\boldsymbol{H}_i}$ denotes the position vector of the i-th individual in the low-dimensional space, $\vec{\boldsymbol{\Delta}}$ denotes the L-dimensional Gaussian perturbation vector, where $\Delta_d = N(0,\sigma)$, $1 \le d \le L$. The perturbation component in each dimension is randomly sampled from a region with a mean of 0 and a variance of σ^2 , as per literature [12]. Here, σ is set to 0.2 for enhanced performance across most test functions. From a geometric perspective, the algorithm generates new individuals within a Gaussian hypersphere centered at the current individual's position.

C. Evolutionary Strategies Employing Multi-Level Hierarchy in High-Dimensional Spaces

In the process of evolution within a high-dimensional space, individuals typically exist in various evolutionary states and possess different potentials for exploring and developing the search space. To distinguish them, they are categorized into distinct hierarchical levels based on their fitness values. Specifically, assuming that *NP* individuals are divided into *NL* hierarchical levels denoted as L_i $(1 \le i \le NL)$, prior to classification, the particles in the population are arranged in ascending order of fitness. Particles with superior fitness belong to higher hierarchical levels, where a lower index indicates a higher level. Consequently, L_1 represents the highest hierarchical level while L_{NL} denotes the lowest one. Each hierarchical level consists of an equal number of individuals referred to as "level size", denoted by LS .

Fig. 3 illustrates an optimization framework based on the Level Learning (LL) strategy. The algorithm arranges the particles in the swarm by sorting them in ascending order of their fitness values and then categorizes them into 4 levels based on their performance. Particles in level 4 learn from individuals in levels L1 to L3, individuals in level 3 learn from individuals in levels L1 and L2, and individuals in level 2 learn from particles in level L1. To safeguard superior particles from being erroneously updated, individuals at level L1 abstain from updating and proceed directly to the next generation. The following presents the update formula for high-dimensional space particles:

$$v_{i,j}^d \leftarrow r_1 v_{i,j}^d + r_2 \left(x_{rl1,k1}^d - x_{i,j}^d \right) + \phi r_3 \left(x_{rl2,k2}^d - x_{i,j}^d \right) \tag{15}$$

$$x_{i,j}^d \leftarrow x_{i,j}^d + v_{i,j}^d \tag{16}$$

In (15) and (16), $\boldsymbol{X}_{i,j} = \left[x_{i,j}^1, ..., x_{i,j}^d, ..., x_{i,j}^M \right]$ denotes the spatial coordinates of the j-th individual within layer L_i of the i-th hierarchy, while M represents the dimensionality of the high-dimensional space. $\boldsymbol{V}_{i,j} = \left[v_{i,j}^1, ..., v_{i,j}^d, ..., v_{i,j}^M \right]$ signifies the particle velocity. Here, $\boldsymbol{X}_{rl1,k1}$ and $\boldsymbol{X}_{rl2,k2}$ represent the positions of two individuals randomly selected from hierarchies $rl1$ and $rl2$, with corresponding indices $k1$ and $k2$ independently chosen from $[1, LS]$. Additionally, $rl1$ and $rl2$ are indices randomly selected from $[1, i-1]$. The variables r_1 , r_2 and r_3 are uniformly distributed random numbers in the range $[0,1]$, while ϕ is a control parameter that governs the influence of a secondary learning objective within the range $[0,1]$. It should be noted that $rl1 < rl2 < i$ implies that $\boldsymbol{X}_{rl1,k1}$ is superior to $\boldsymbol{X}_{rl2,k2}$, both are superior to $\boldsymbol{X}_{i,j}$.

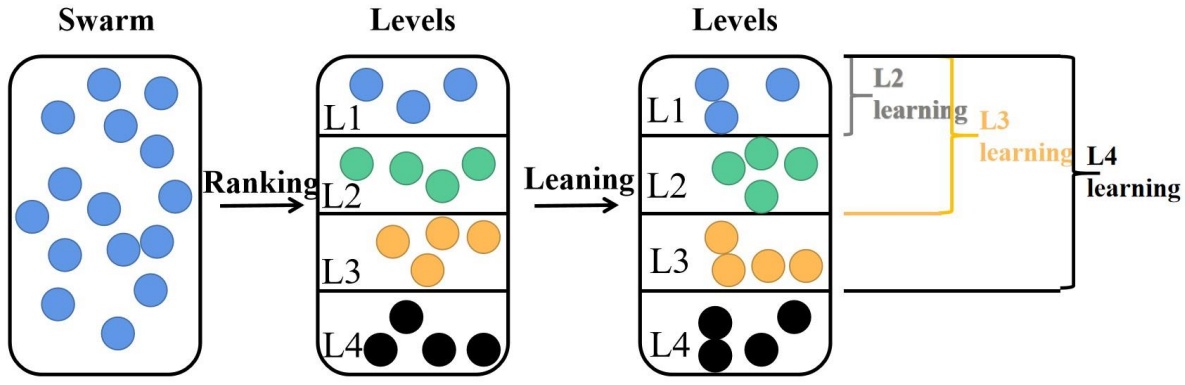

Fig. 3. Based on level learning strategy (LL).

D. The Flow of DRSAE

Fig. 2 illustrates the flowchart of DRSAE, while Algorithm I delineates the procedural steps of DRSAE.

Algorithm I: DRSAE Algorithm

Input: population P, maximum number of fitness evaluations FE_{max}, dimension of the problem D.
Output: The final solution x and its fitness $f(x)$.

```
FE_cur = 0;
Establish database of {position, its fitness};
Initialize the swarm randomly and calculate the fitness of particles;
Initialize surrogate model RBF using first generation population;
Update FE_cur;
While size(database) < 150 do
    P' = DE(P);
    Evaluate the fitness of P': f(P');
    Select and update P and f(P);
    database = {database, new particles};
    If new data added to database then
        RBF = UpdateRBF(new_data);
    End if
End while
While FE_cur < FE_max do
    Population split: Split P into P1 and P2 according to the dynamic size adjustment strategy;
    /* high dimensional evolution */
    P1' = LL(P1) by (15)—(16);
    Evaluate the fitness of P1': f(P1');
    Select and update P1 and f(P1);
    /* low dimensional evolution */
    Dimensionality reduction: P2_low = ELM-AE(P2);
    P2_low' = Mutation (P2_low) by (14);
    Reconstruction: P2' = ELM-AE(P2_low);
    Evaluate the fitness of P2': f(P2');
    Select and update P2 and f(P2);
    Update database and FE_cur;
    If new data added to database then
        RBF = UpdateRBF(new_data);
    End if
End while
```

E. Complexity Analysis of DRSAE

In the initial iteration of the algorithm, the time complexity for collecting the initial training samples for the surrogate model is denoted as $O(KD)$, where K represents the number of training samples activated by the surrogate model. In this study, we assume K to be $5d$, resulting in a time complexity of $O(D^2)$ for this stage. During each iteration process, sorting the subpopulation (i.e., ranking) has a time complexity of $O(N\log_2(N))$. The maintenance of the surrogate model carries a time complexity of $O(D(N/2)^5)$, which can be approximated as $O(DN^5)$. Within high-dimensional space during hierarchical learning evolution, lower-level particles learn from higher-level particles; according to (15), this results in a time complexity of $O(ND)$. For Gaussian variation evolution based on dimensionality reduction in low-dimensional space as per (6), both dimensionality reduction and restoration have a time complexity of $O(DN^2)$; and according to (14), the variation process has a time complexity of approximately $O(D^2+DN^5+ND+DN^2)$. Consequently, ultimately, it is determined that DRSAE algorithm exhibits a time complexity of approximately $O(D^2+DN^5)$.

IV. Experimental Results and Discussion

To assess the performance of DRSAE, in this session, we conducted experiments on seven benchmark functions [13-17] commonly employed in the field. These functions demonstrate varying decision space dimensions and function characteristics, offering a comprehensive illustration of DRSAE's applicability to HEPs. The test functions ranged in dimensionality from 50 to 200 dimensions, with essential information about these benchmark functions presented in Table I. Each algorithm was independently executed on each benchmark function 20 times in all experiments, and the results were subsequently averaged. The best average value for each benchmark function is highlighted in bold. The experimental environment consisted of an Intel i5 CPU running at 2.50 GHz, equipped with 8 GB RAM and operating on Windows 10 system alongside matlab R2020a.

TABLE I. Information Regarding the Benchmark Function

Fun	Name	Design space	f^* †	Property
F1	Ellipsoid	$[-5,5]^d$	0	Unimodal
F2	Rosenbrock	$[-2,2]^d$	0	Multimodal with narrow valley
F3	Ackley	$[-32,32]^d$	0	Multimodal
F4	Griewank	$[-600,600]^d$	0	Multimodal
F5	Rastrigin	$[-5,5]^d$	0	Multimodal
F6	Shifted rotated F5	$[-5,5]^d$	-330	Multimodal & Complex
F7	Hybrid ‡ function	$[-5,5]^d$	10	Multimodal & Complex

†: means global optimum.
‡: Rotated hybrid composition function with a narrow basin for the global optimum.

A. Algorithm Peers and Parameter Setting

DRSAE integrates the concept of agent models. To assess the efficacy of DRSAE, this study assesses six agent-based optimization algorithms, namely GSGA [13], SA_COSO [14], ESAO [15], SHPSO [16], SAMSO [17], and SAEO [7]. Similar to traditional algorithms, the population size in DRSAE is fixed at 100, with a maximum number of re-evaluations per generation set at 5. The activation criteria for the agent models are contingent on problem dimensionality and are triggered when the training point count reaches 1000. Experimental findings demonstrate that DRSAE requires fewer computational resources than the algorithm proposed in [18] while approaching closer to an optimal solution. This unequivocally substantiates the substantial advantages of DRSAE in terms of optimization efficiency and effectiveness.

B. Sensitivity Analysis of Surrogate Model Activation Point

Surrogate models have demonstrated high efficacy in addressing HEPs. However, a critical concern arises: when should the surrogate model be initiated within this framework? The initial training of the surrogate using data samples can ensure its

accuracy while managing time resource overhead. We designated different time points for activating the surrogate model, consuming 400, 600, and 800 FEs respectively. We evaluated the performance of each parameter using F1 and F5 metrics, highlighting the optimal result for each function. The optimization outcomes are presented in Table II, and the convergence curve is depicted in Fig. 4.

Fig. 4 illustrates that the activation strategy for data surpasses other strategies in identifying an optimal solution with limited resources, indicating that the dimension-based strategy is well-suited for activating the agent model. As shown in Fig. 4(a)-(b), it is apparent that even if the constructed agent is sufficiently precise, delaying its activation until later stages of the optimization process may restrict subsequent optimization due to the limited number of remaining FEs. For high-dimensional problems, as demonstrated in Fig. 4(c)-(d), an agent model built with a small number of data samples does not offer accurate search guidance. Therefore, it can be inferred that there is no discernible efficiency advantage to activating the agent early in the optimization process, and using data samples as an activation condition yields optimal results.

C. Sensitivity Analysis of Dynamically Adjusting Subpopulation Size

To assess the efficacy of dynamically adjusting subpopulation size, we compared the dynamic subpopulation size adjustment strategy with the fixed subpopulation size strategy using DRSAE (Z1, Z2) to present experimental findings. The study evaluated F1 and F5 functions, each possessing distinct characteristics, with bold highlighting the optimal results for each function.

As depicted in Table III, the dynamic subpopulation size adjustment strategy demonstrates superiority over the fixed subpopulation size strategy throughout the optimization process. Fig. 5 illustrates the dynamic sizing of subpopulation I for the F1 function at dimensions 50 and 200. Given that subpopulation I primarily emphasizes exploration, it is initially larger and gradually diminishes according to (10) as optimization progresses to emphasize exploitation. With an increase in search space dimensionality, a more intricate environment necessitates efficient exploration during initial stages; hence, as problem dimensions escalate, so does the initial size of subpopulation I.

D. Comparative Experiments on Benchmark Functions

Table IV presents the mean and standard deviation of DRSAE's independent runs conducted 20 times on 7 test functions listed in Table I. The convergence curves from the original papers of SA_COSO [14], SAEO [7], SHPSO [17], and ESAO [15] were re-plotted, and the Wilcoxon signed-rank test results were calculated at a significance level of $\alpha = 0.05$. The optimal result for each function is indicated in bold typeface. "Standard deviation" is shortened to "Std devi" for compact formatting.

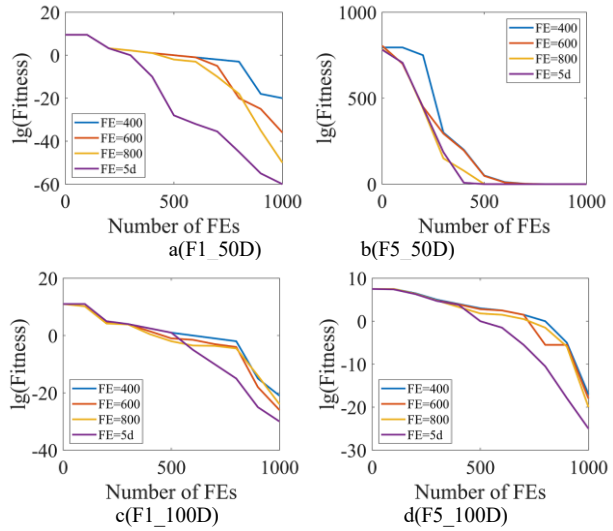

Fig. 4. Effects of activating surrogate models time-point.

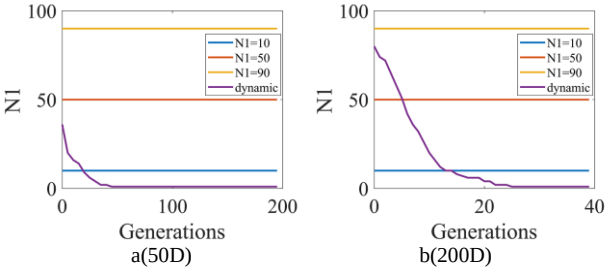

Fig. 5. The dynamic size of sub-population.

TABLE II. OPTIMIZATION RESULTS OF ACTIVATING THE AGENT MODEL AT DIFFERENT TIME POINTS

Fun & Dimension	Metrics	400	600	800	5d
F1(50)	**Mean**	4.86e-23	1.77e-17	1.09e-09	**1.51e-26**
	Std deviation	2.12e-22	3.01e-16	2.18e-08	**2.46e-26**
F5(50)	**Mean**	**0.00e-00**	4.00e-11	6.10e-10	**0.00e-00**
	Std deviation	**0.00e-00**	1.60e-10	4.08e-09	**0.00e-00**
F1(100)	**Mean**	1.29e-10	1.42e-12	5.43e-10	**8.99e-13**
	Std deviation	1.52e-09	1.05e-11	1.52e-09	**3.12e-12**
F5(100)	**Mean**	1.37e-08	1.73e-09	1.12e-08	**2.22e-11**
	Std deviation	2.69e-07	1.84e-08	2.63e-07	**8.87e-11**
F1(200)	**Mean**	4.54e-03	1.18e-02	6.92e-04	**1.79e-04**
	Std deviation	1.06e-02	2.94e-02	9.51e-04	**2.45e-04**
F5(200)	**Mean**	6.63e-01	1.56e-02	9.32e-03	**2.26e-04**
	Std deviation	2.04e-00	2.40e-02	3.24e-02	**5.15e-04**

TABLE III. OPTIMIZATION RESULTS OF DIFFERENT SUBPOPULATION ALLOCATION STRATEGIES

Fun & Dimension	Metrics	DRSAE(10,90)	DRSAE(50,50)	DRSAE(90,10)	DRSAE(dynamic)
F1(50)	**Mean**	3.51e-02	5.61e-02	1.08e-02	**1.51e-26**
	Std deviation	4.78e-02	9.67e-02	6.70e-03	**2.46e-26**
F5(50)	**Mean**	1.05e+01	1.05e+01	3.64e-01	**0.00e-00**
	Std deviation	7.99e+00	1.95e+01	5.90e-01	**0.00e-00**
F1(100)	**Mean**	2.48e-02	1.84e-02	3.21e-02	**8.99e-13**
	Std deviation	4.82e-02	3.09e-02	2.09e-02	**3.12e-12**
F5(100)	**Mean**	2.28e+00	8.98e-01	1.77e-01	**2.22e-11**
	Std deviation	4.37e+00	2.57e+00	8.26e-02	**8.87e-11**
F1(200)	**Mean**	8.24e-04	5.62e-03	4.35e-02	**1.79e-04**
	Std deviation	2.41e-04	2.04e-03	1.04e-02	**2.45e-04**
F5(200)	**Mean**	1.54e-03	1.24e-02	3.41e-01	**2.26e-04**
	Std deviation	1.08e-03	1.17e-02	2.58e-01	**5.15e-04**

TABLE IV. COMPARISON RESULTS OF 7 ALGORITHMS ON TEST FUNCTIONS F1-F7 ACROSS DIMENSIONS 50 TO 20

Fun	Dim	Metrics	SA_COSO	SHPSO	ESAO	SAMSO	GSGA	SAEO	DRSAE
F1	50	**Mean**	4.66e+01(+)	7.17e+00(+)	7.40e-01(+)	5.13e-01(+)	6.21e-01(+)	1.51e-26(≈)	**1.21e-27**
		Std devi	1.74e+01	2.42e+00	5.55e-01	2.85e-01	4.84e-01	2.46e-26	**2.46e-22**
	100	**Mean**	1.03e+03(+)	7.61e+01(+)	1.28e+03(+)	7.21e+01(+)	1.23e+01(+)	**8.99e-13(≈)**	5.99e-12
		Std devi	3.17e+02	2.14e+01	1.34e+02	1.78e+01	9.39e+00	**3.12e-12**	6.22e-11
	200	**Mean**	1.63e+04(+)	2.35e+03(+)	1.76e+04(+)	1.52e+03(+)	3.14e+03(+)	1.79e-04(+)	**3.70e-05**
		Std devi	2.98e+03	3.25e+03	1.17e+03	2.12e+02	6.14e+03	2.45e-04	**1.23e-04**
F2	50	**Mean**	2.53e+02(+)	5.13e+01(+)	**4.74e-02(-)**	5.01e+01(+)	4.82e+01(+)	4.89e+01(+)	4.83e-02
		Std devi	5.67e+01	2.00e+00	**1.71e+00**	7.68e-01	7.66e-01	1.66e-02	4.16e+00
	100	**Mean**	2.71e+03(+)	1.65e+02(+)	5.79e+02(+)	2.86e+02(+)	1.09e+02(+)	9.88e+01(≈)	**7.62e+01**
		Std devi	1.17e+02	2.63e+01	4.48e+01	5.25e+01	1.17e+01	2.83e-02	**2.41e-02**
	200	**Mean**	1.64e+04(+)	2.48e+03(+)	4.31e+03(+)	1.15e+03(+)	4.58e+02(+)	1.98e+02(+)	**1.02e+02**
		Std devi	4.09e+03	1.96e+02	2.84e+02	1.16e+02	1.16e+02	5.43e-02	**1.36e-02**
F3	50	**Mean**	8.86e+00(+)	2.60e+00(+)	1.43e+00(+)	1.53e+00(+)	2.16e-02(+)	8.55e-14(≈)	**2.56e-15**
		Std devi	1.10e+00	2.48e-01	2.49e-01	4.36e-01	2.37e-02	4.42e-13	**3.51e-13**
	100	**Mean**	1.57e+01(+)	4.11e+00(+)	1.04e+01(+)	6.12e+00(+)	1.31e+00(+)	6.96e-07(≈)	**5.12e-07**
		Std devi	5.02e-01	5.92e-01	2.11e-01	4.09e-01	9.68e-01	1.05e-06	**3.25e-06**
	200	**Mean**	1.78e+01(+)	2.13e+01(+)	1.46e+01(+)	1.20e+01(+)	2.20e+01(+)	1.86e-03(+)	**2.14e-04**
		Std devi	2.23e-02	1.02e-01	2.19e-01	4.00e-01	6.20e-01	1.22e-03	**1.94e-03**
F4	50	**Mean**	5.63e+00(+)	9.45e-01(+)	9.40e-01(+)	6.66e-01(+)	3.46e-01(+)	**6.92e-15(≈)**	8.65e-15
		Std devi	8.92e-01	5.39e-02	4.21e-02	1.07e-01	7.15e-02	**8.85e-14**	6.11e-14
	100	**Mean**	6.33e+01(+)	1.07e+00(+)	5.73e+01(+)	1.06e+00(+)	7.06e-01(+)	**1.39e-07(-)**	8.65e-06
		Std devi	1.90e+01	2.04e-02	5.84e+00	2.64e-02	7.06e-02	**8.79e-06**	1.25e-05
	200	**Mean**	5.77e+02(+)	3.14e+02(+)	5.72e+02(+)	9.03e+00(+)	1.03e+01(+)	3.79e-02(+)	**2.85e-03**
		Std devi	1.01e+02	6.58e+01	3.60e+01	1.33e+00	1.69e+00	3.16e-06	**6.32e-03**
F5	50	**Mean**	3.22e+02(+)	3.89e+02(+)	4.21e+02(+)	3.77e+01(+)	1.53e+00(+)	0.00e-00(-)	**15.99e-12**
		Std devi	3.83e+01	6.27e+01	1.24e+01	7.82e+00	4.36e-01	0.00e-00	**6.22e-11**
	100	**Mean**	8.81e+02(+)	8.78e+02(+)	4.15e+02(+)	4.29e+02(+)	6.55e+02(+)	2.22e-11(≈)	**6.15e-12**
		Std devi	7.01e+01	8.93e+01	6.73e+01	5.72e+01	6.31e+01	8.87e-11	**6.24e-11**
	200	**Mean**	5.77e+02(+)	5.72e+02(+)	3.14e+02(+)	9.03e+00(+)	1.03e+01(+)	3.79e-02(+)	**2.85e-03**
		Std devi	1.01e+02	3.60e+01	6.58e+01	1.33e+00	1.69e+00	3.16e-06	**6.32e-03**
F6	50	**Mean**	2.35e+02(−)	1.22e+02(−)	1.99e+02(−)	8.69e+02(+)	**7.58+01(−)**	7.71e+02(+)	6.66e+02
		Std devi	4.09e+01	2.59e+01	4.58e+01	3.17e+01	**4.99e+01**	5.89e+01	6.70e+01
	100	**Mean**	1.27e+03(+)	8.01e+02(−)	7.13e+02(−)	7.37e+02(−)	**6.72e+02(−)**	2.02e+03(+)	1.85e+03
		Std devi	1.17e+02	7.22e+01	2.65e+01	4.20e+01	**2.97e+01**	1.34e+02	9.16e+01
	200	**Mean**	3.92e+03(−)	4.15e+03(−)	5.38e+03(+)	4.96e+03(+)	**2.56e+03(−)**	4.80e+03(≈)	4.73e+03
		Std devi	2.72e+02	2.98e+02	1.56e+02	1.38e+02	**2.68e+02**	2.19e+02	2.89e+02
F7	50	**Mean**	1.08e+03(+)	1.00e+03(+)	9.75e+02(+)	9.70e+02(+)	9.70e+02(+)	9.10e+02(≈)	9.10e+02
		Std devi	3.66e+01	2.12e+01	3.71e+01	2.92e+01	1.81e+01	4.71e-03	**0.00e+00**
	100	**Mean**	1.36e+03(+)	1.41e+03(+)	1.37e+03(+)	1.29e+03(+)	1.25e+03(+)	9.10e+02(≈)	**9.10e+02**
		Std devi	3.08e+01	3.82e+01	2.75e+01	3.34e+01	2.45e+01	1.39e-02	**1.65e-09**
	200	**Mean**	1.34e+03(+)	6.27e+02(+)	1.45e+03(+)	1.34e+03(+)	5.25e+03(+)	9.10e+02(+)	**4.16e+01**
		Std devi	2.46e+01	1.15e+01	2.04e+01	2.43e+01	1.87e+01	5.72e-04	**2.79e-04**

Table IV demonstrates that DRSAE surpasses most comparable algorithms in efficiently identifying the vicinity of the optimal solution within a limited number of evaluations, while many others fall short of reaching the optimum. Notably, DRSAE exhibits strong performance on F5 with a minimum value characteristic indicative of a regular distribution. Among these, SAEO [7] shows relatively superior performance due to its utilization of dimensionality reduction. Furthermore, the ELM-AE dimensionality reduction network employed by DRSAE accurately restores high-dimensional space without information loss, resulting in overall better performance compared to SAEO. ESAO showcases commendable exploration ability owing to differential evolution, leading to superior performance on F2 with 50 dimensions compared to DRSAE.

Plot convergence curves for the 100D and 200D problems, as illustrated in Fig. 6 and Fig. 7, respectively. Analysis of figures indicates that DRSAE demonstrates faster convergence to the optimal solution compared to other surrogate-assisted algorithms such as SAEO. With increasing dimensionality, DRSAE exhibits superior optimization speed relative to other algorithms due to its enhanced capability in identifying promising regions within low-dimensional space. GSGA [13] performs exceptionally well on F6, a complex multimodal function, by iteratively employing alternative functions at the cost of computational burden to enhance opportunities for finding the optimal solution. The suboptimal performance of DRSAE on F6 is primarily attributed to the inability of dimensionality reduction in simplifying the complexity of F6's landscape. Overall, DRSAE ranks first in average for both optimization speed and convergence performance, thus fully demonstrating its superiority. Upon activation of the surrogate model, fitness value sharply decreases within the same range of function evaluations.

E. Effectiveness Analysis of Dimensionality Reduction Strategies

To assess the efficacy of incorporating a low-dimensional search space, this study selects the F1 and F5 functions with dimensions ranging from 50 to 200 for evaluating both DRSAE and its non-low-dimensional version, DRSAE*. The optimization results for DRSAE and DRSAE* are presented in Table V.

Drawing upon the findings presented in Table V, it is evident that ELM-AE has significantly augmented the search efficiency of DRSAE* while maintaining an equivalent number of FEs. The results signify a noteworthy reduction in the average optimal values, leading to expedited optimization speed for DRSAE. ELM-AE effectively steers the exploration of low-dimensional search space, integrates high-dimensional information, and extracts latent features, thereby facilitating rapid identification of promising regions and substantial reduction in unnecessary FE consumption, particularly for problems with potential convergence issues. For problems up to 200 dimensions, leveraging the low-dimensional search space in ELM-AE has widened the gap between optimal values, demonstrating its advantage in precise and efficient dimensionality reduction.

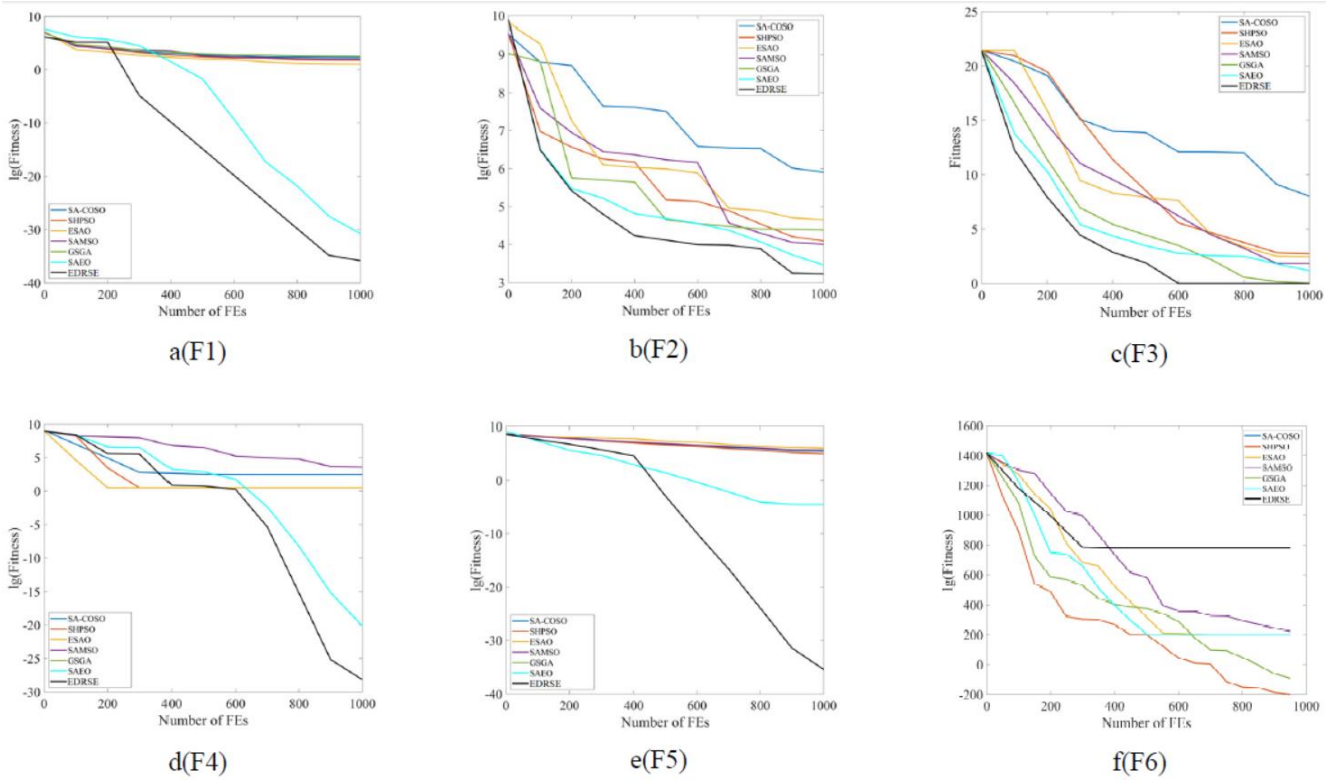

Fig. 6. Convergence curves of different algorithms on 100D benchmark problems.

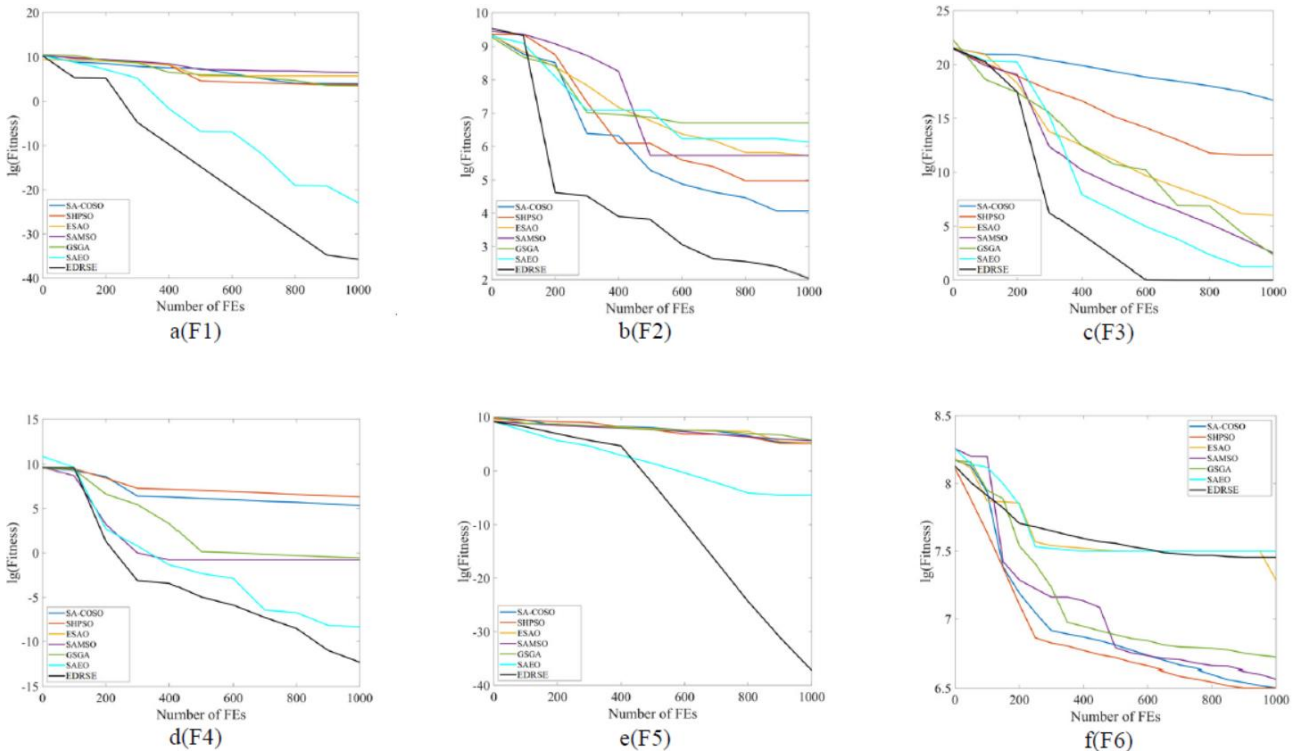

Fig. 7. Convergence curves of different algorithms on 200D benchmark problems.

TABLE V. OPTIMIZATION RESULTS OF DRSAE* AND DRSAE

Fun & Dimension	Metrics	DRSAE*	DRSAE
F1(50)	**Mean**	4.18e-03	**1.51e-26**
	Std deviation	1.49e-03	**2.46e-26**
F5(50)	**Mean**	8.06e-02	**0.00e+00**
	Std deviation	7.11e-02	**0.00e+00**
F1(100)	**Mean**	1.55e-02	**8.99e-13**
	Std deviation	7.42e-03	**3.12e-12**
F5(100)	**Mean**	8.51e-02	**2.22e-11**
	Std deviation	4.53e+00	**8.87e-11**
F1(200)	**Mean**	1.75e+01	**1.79e-04**
	Std deviation	1.52e+01	**2.45e-04**
F5(200)	**Mean**	8.81e+02	**2.26e-04**
	Std deviation	5.14e+02	**5.15e-04**

V. CONCLUSION

This paper introduces an effective dimensionality reduction assisted evolutionary framework (DRSAE) for addressing high-dimensional expensive evaluation function problems (HEPs). The primary challenge in HEPs lies in the high cost of evaluating the function, necessitating the rapid identification of optimal solutions within a limited number of function evaluations. The algorithm makes two key contributions: 1) Incorporating an efficient and accurate low-dimensional search space into the traditional agent-based algorithm, enabling precise reduction and reconstruction of high-dimensional space to minimize position information errors resulting from switching between spaces and expedite the discovery of a more promising solution space; 2) Implementing hierarchical learning of particles in high-dimensional space, allowing lower-level particles to learn from superior ones to enhance population diversity. Furthermore, specific activation conditions for the agent model are established for different dimensional problems.

In order to assess the performance of DRSAE, it was compared with other established algorithms across seven commonly utilized functions. The experimental results indicate that in most cases, DRSAE performs admirably, with SAEO demonstrating relatively superior performance. This can be attributed to the fact that SAEO also operates within a low-dimensional search space; however, its low-dimensional model is based on AE. In contrast to DRSAE, where ELM-AE produces a smaller reconstruction error from the low-dimensional space to the high-dimensional space. As a result, DRSAE is able to more accurately reconstruct features in the high-dimensional space and achieve improved performance. In addition, better results may be achieved by improving the ELM network structure to stack ELM hidden layers.

VI. FUTURE WORK

We endeavor to integrate state-of-the-art Evolutionary Algorithms (EAs) [19-23] into the DRSAE in order to tackle High-dimensional Expensive Problems (HEPs), while exploring the underlying theory in future research. It is worth noting that this framework has the potential for extension to address multi-objective optimization problems, dynamic optimization problems, and constrained optimization problems, thereby validating the effectiveness of DRSAE in relevant real-world scenarios and expanding its applicability in high-dimensional expensive optimization domains.

REFERENCES

[1] Hu Rong, Chen Wenbo, Qian Bin, Guo Ning, and Xiang Fenghong. "Learning more than ant colony algorithm to solve the green yard vehicle routing problem". *Journal of System Simulation*, China,33(09):2095-2108, 2021.

[2] Li Han, Du Peng, Du Ying, et al. "Multi-path routing selection method for wireless body area networks based on genetic algorithm". *Journal of Jilin University (Engineering and Technology Edition)*, China, 52(11): 2706-2711, 2022.

[3] Yang Hua. "Research on feature gene selection method based on particle swarm algorithm." *Hunan University*, China, 2010.

[4] Xu Kangyu, Liu Yuan, Li Miqing, Yang Shengxiang, Zou Juan, and Zheng Jinhua. "A review of evolutionary high-dimensional multi-objective optimization." *Control Engineering*, China, 30(08): 1436-1449, 2023.

[5] Zhao Yuxiang, Li Qiang, Liu Ziyu. "Application of gaussian process surrogate Model in Large-Scale Global Optimization." *Control and Decision*, China, 36(3): 577-583, 2021.

[6] C. Sheng, J. Hundley. "Data-dimensionality reduction. " *Whitman College*,https://www.whiman.edu/Docments/Acadeics/Mathematics/2019/Sheng-Hundley.pdf. USA, 2019.

[7] M. Cui, L. Li, M. Zhou and A. Abusorrah, "Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 676-689, Aug. 2022.

[8] F. Li, X. Cai, L. Gao and W. Shen, "A Surrogate-Assisted Multiswarm Optimization Algorithm for High-Dimensional Computationally Expensive Problems," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1390-1402,2020.

[9] Huang Guangbin, Song Shiji, and You Kai. "Trends in extreme learning machines." *Neural Networks*, 61: 32-48, 2015.

[10] Zhang Ting and Yu Li. "Extreme learning machine: algorithms and applications." *Neural Computing and Applications*, 32(11):7041-7059, 2020.

[11] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326-347, June 2014.

[12] R. Mendes and A. S. Mohais, "DynDE: a differential evolution for dynamic optimization problems," *IEEE Congress on Evolutionary Computation*, Edinburgh, UK, pp. 2808-2815 Vol. 3, 2005.

[13] X. Cai, L. Gao and X. Li, "Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 365-379, April 2020.

[14] C. Sun, Y. Jin, R. Cheng, J. Ding and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644-660, Aug. 2017.

[15] X. Wang, G. G. Wang, B. Song, P. Wang and Y. Wang, "A novel evolutionary sampling assisted optimization method for high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 815-827, Oct. 2019.

[16] Haibo Yu, Ying Tan, Jianchao Zeng, Chaoli Sun, and Yaochu Jin. "Surrogate-assisted hierarchical particle swarm optimization," *Information Sciences*, 454(2):59–72. 2018.

[17] F. Li, X. Cai, L. Gao and W. Shen, "A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1390-1402, March 2021.

[18] Sun, C., Ding, J., and Zeng, J. "A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems. "*Memetic Comp.*10(2), 123–134, 2018.

[19] J. Zhang, P. Wen and A. Xiong, " Application of improved quantum particle swarm optimization algorithm to multi-task assignment for heterogeneous UAVs,"*2022 6th Asian Conference on Artificial Intelligence Technology (ACAIT)*, Changzhou, China, pp. 1-5, 2022.

[20] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 904-916, July 2019.

[21] Y. Yu, S. Gao, Y. Wang and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 379-394, March 2019.

[22] W. Qingling and J. Yubo, "Research on improved particle swarm optimization algorithm based on simulated annealing algorithm,"*2023 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, Ottawa, ON, Canada, pp. 360-362,2023.

[23] T. Wang, W. Miao and Z. Zeng, "Optimization method of Data interaction in power IoT based on particle swarm algorithm," *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, Guangzhou, China, pp. 351-355, 2022.