Journal of Computer and Communications, 2026, 14(2), 64-82
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Research on Adaptive Surrogate-Assisted Evolutionary Algorithm Based on SISSO Model Selector

Baolei Li[1,2], Weizhong Qiu[1], Chunying Chen[3], Xiang Liu[1], Ziwei Zhao[1], Wentao You[1]

[1]School of Mathematics and Computer Science, Gannan Normal University, Ganzhou, China
[2]School of Intelligent Manufacturing and Future Energy, Gannan Normal University, Ganzhou, China
[3]Shiyuan Central Primary School, Ganxian District, Ganzhou, China
Email: libaolei@gnnu.edu.cn

How to cite this paper: Li, B.L., Qiu, W.Z., Chen, C.Y., Liu, X., Zhao, Z.W. and You, W.T. (2026) Research on Adaptive Surrogate-Assisted Evolutionary Algorithm Based on SISSO Model Selector. *Journal of Computer and Communications*, **14**, 64-82.
https://doi.org/10.4236/jcc.2026.142004

Received: January 20, 2026
Accepted: February 11, 2026
Published: February 14, 2026

Abstract

Surrogate-assisted evolutionary algorithms are widely used to solve expensive optimization problems due to their high search efficiency. However, a single model struggles to fit various fitness landscapes with different characteristics. How to adaptively select the most suitable model based on the problem's fitness landscape characteristics remains an open research area. Therefore, this paper proposes an adaptive surrogate model selection framework based on an SISSO model selector. This method extracts problem features from the optimization process and dynamically selects the surrogate model most suitable for the current stage via the SISSO model selector, thereby enhancing the algorithm's adaptability to different function characteristics. Experiments on CEC2022 test functions show that the proposed method achieves significantly lower mean errors and stable variances on most functions. Particularly on functions F10 and F11, the SISSO method reduces the mean error by more than an order of magnitude compared to methods like ElasticNet, Lars, Lasso, and Ridge, demonstrating excellent comprehensive performance.

Keywords

SISSO, Evolutionary Algorithm, Adaptive, Features

1. Introduction

In solving complex optimization problems, Evolutionary Algorithms (EAs) are widely used in material prediction fields due to their strong global search capability and black-box nature concerning the objective function [1] [2]. They explore and exploit the solution space by simulating biological evolution mecha-

nisms such as mutation, crossover, and selection [3]. However, with increasing computational complexity and problem dimensionality, traditional EAs face two major challenges: high computational cost and declining local search ability. Surrogate-Assisted Evolutionary Algorithms (SAEAs) improve search efficiency and solution quality by constructing approximate models of the objective function (e.g., Gaussian Process, Random Forest) to replace some expensive real evaluations, thereby reducing the number of costly real function calls [4]. Nevertheless, existing SAEAs mostly adopt static surrogate model strategies, making it difficult to adapt to the dynamic changes in objective function characteristics during optimization [5]. For example, in the early stages of hybrid objective function optimization, Random Forest might excel at capturing nonlinear relationships, but in later fine-tuning stages, Gaussian Process may have an advantage due to its uncertainty assessment capability [6] [7]. To address these issues, this paper proposes an adaptive surrogate-assisted evolutionary algorithm framework based on an SISSO model selector (Adaptive Surrogate-Assisted Evolutionary Algorithm with SISSO Classifier), named SISSO-ASEA. Its core contributions are as follows:

1) An adaptive selection framework based on an SISSO model selector is proposed, enabling automatic surrogate model selection.

2) Based on extensive experiments, the high efficiency and effectiveness of the proposed algorithm are demonstrated.

2. Related Work

2.1. Common Surrogate Models

SAEAs are a framework for evolutionary algorithms that has garnered significant attention in recent years and has become a research hotspot for Expensive Multi-objective Optimization Problems (EMOPs) [8]. Commonly used surrogate models include:

1) Gaussian Process Regression (GPR) [9]: It models the probability distribution of the objective function through kernel functions, with the output mean and covariance given by Equation (1).

$$\mu(x) = k^{\mathrm{T}}(x, X)K^{-1}y,\ \sigma^2(x) = k(x, x) - k^{\mathrm{T}}(x, X)K^{-1}k(x, X) \tag{1}$$

where K^{T} is the training data covariance matrix, and $k(x, X)$ is the kernel function vector between the test point x and the training set X.

2) Random Forest (RF) [10]: Random Forest predicts results through decision trees, with its output being the mean of all trees as shown in Equation (2).

$$\hat{f}(x) = \frac{1}{M}\sum_{i=1}^{m} T_i(x) \tag{2}$$

where T_i is the predicted value of the i-th tree, and M is the number of trees.

3) Support Vector Regression (SVR) [11]: SVR is trained by maximizing the margin and minimizing prediction errors, with its objective function given by Equation (3).

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n}\left(\varepsilon_i + \varepsilon_i^*\right) \\ &\text{s.t. } y_i - \left(w^{\mathrm{T}}\phi\left(x_i\right) - b\right) \le \varepsilon + \xi_i \\ &\quad w^{\mathrm{T}}\left(x_i\right) + b - y_i \le \varepsilon + \xi \end{aligned} \tag{3}$$

where $\phi\left(x_i\right)$ is the kernel mapping function, ε is the tolerance error, and ξ is the slack variable.

2.2. Heterogeneous Surrogate Models

To compare the tendencies and limitations of different heterogeneous surrogate models, based on Table 1, this paper statistically summarizes the tendency characteristics of Gaussian Process Regression, Random Forest, Support Vector Regression, etc., in terms of problem dimensionality, smoothness, complexity, etc., as listed in Table 1.

Table 1. Applicability analysis of heterogeneous surrogate models.

Model	Inclined Problem Characteristics	Limiting Problem Characteristics
Gaussian Process Regression	Highly smooth objective function.	Computational cost increases significantly with higher problem dimensions.
Random Forest	Objective function exhibits strong nonlinearity.	Predictions may be unreliable near the boundaries of the search space.
Support Vector Regression	Objective function may contain data noise	Training time can be long.
LightGBM	Complex nonlinear objective function.	Prone to overfitting with small datasets.
Bayesian Ridge	Objective function has high smoothness index	Limited ability to fit complex functional relationships.

3. Adaptive Selection Algorithm Based on SISSO

3.1. Algorithm Workflow

The overall workflow of the proposed adaptive surrogate model selection framework based on the SISSO model selector is shown in Figure 1.

Step 1: Determine basic function characteristics. Select 15 representative basic functions (F1 - F15) from the optimization function library, covering different types such as unimodal, multimodal, hybrid, and composite. For each basic function, perform uniform random sampling, with the sample size gradually increasing from 10 to 200. For the sampled data under each sample size, compute eight feature values: mean of objective values, standard deviation, skewness and kurtosis, correlation strength, nonlinear index, smoothness index, and ruggedness index. Observe the trend of each feature value for each basic function as the sample size increases, plot the feature-value vs. sample-size change curve, and analyze the convergence of the feature values. The feature value at convergence is taken as the function's characteristic value.

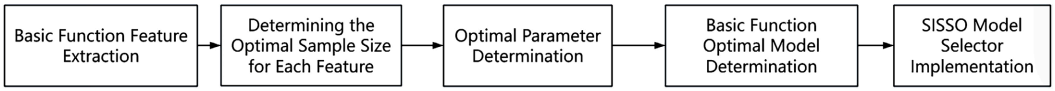

Figure 1. Overall workflow of adaptive surrogate model selection framework based on sisso model selector.

Step 2: Determine the optimal model for basic functions. First, select 12 candidate surrogate models (M1 - M12). For each basic function, use the RandomizedSearchCV() method from the sklearn library to optimize model parameters within a defined hyperparameter grid, determining the optimal parameters for each model fitting each function. Using the optimized surrogate models, perform five-fold [cross-validation] {.mark} on the 15 basic functions, and count the distribution of core parameters where each model performs best across all functions. Under the optimal parameter configuration, re-evaluate the performance of each model under different sample sizes. The model that does not change when the sample size increases sufficiently is considered the optimal model for the basic function.

Step 3: Implementation of the SISSO model selector. Evaluate the surrogate models' fitting effectiveness on test functions using the R^2 metric, obtaining an R^2 performance matrix for each model on the basic functions. Based on R^2 values, calculate the Euclidean distance between models to construct a model correlation matrix. Build a complete graph with models as nodes and Euclidean distances as edge weights. First, transform the model numbering problem into a Traveling Salesman Problem to find the shortest traversal path. Number the models according to their order on the shortest path. Use the SISSO algorithm to obtain a mapping function from the 8 features to 5 model numbers, implementing model selection.

3.2. Algorithm Implementation

3.2.1. Initial Sampling

Use Symmetric Latin Hypercube Design (SLHD) to generate initial sample points [12].

$$X_{init} = \left\{ x^{(1)}, x^{(2)}, \cdots, x^{(n)} \right\} \subset [l, u]^d \tag{4}$$

where, $x^{(n)}$ is the n-th sample point, l is the lower bound of variables, u is the upper bound of variables, and d is the problem dimensionality.

3.2.2. Feature Extraction Technique

Extract multi-dimensional features from the problem and sampled data. Problem structural features: dimensionality, search space volume, boundary range. Data statistical features: mean of objective values, standard deviation, skewness, kurtosis. Function characteristic features: correlation strength, nonlinear index, smoothness index, ruggedness index [13] [14].

For a given optimization problem, let the variable dimensionality be d, the lower variable bound be l, the upper bound be u, the initial sample points be X, and the objective values be y. Feature extraction can be uniformly expressed as:

$$m = \phi(X, y, l, u, d) \tag{5}$$

where m is the feature vector, and $\phi(\bullet)$ represents the feature extraction operator.

① Problem structural features. Include dimensionality, space volume, boundary range. Define the problem structural feature vector as:

$$m_{struct} = (d, \log V, R) \tag{6}$$

where space volume and boundary range can be expressed as:

$$V = \prod_{i=1}^{d} (u_i - l_i) \tag{7}$$

$$R = \frac{1}{d} \sum_{i=1}^{d} (u_i - l_i) \tag{8}$$

② Sampled data statistical features. Include mean of objective values f1, standard deviation f2, skewness f3, kurtosis f4. Define the data statistical feature vector as:

$$m_{data} = (\bar{y}, \sigma_y, \gamma_1, \gamma_2) \tag{9}$$

where the skewness coefficient and kurtosis coefficient are respectively:

$$\gamma_1 = \frac{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^3}{\sigma_y^3} \tag{10}$$

$$\gamma_2 = \frac{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^4}{\sigma_y^4} \tag{11}$$

③ Function characteristic features. Include correlation strength f5, nonlinear index f6, smoothness index f7. Define the function characteristic feature vector as:

$$m_{func} = (\rho, \eta, \xi) \tag{12}$$

Correlation strength (ρ): This feature measures the degree of linear correlation between input variables, with the formula as follows.

$$\rho = \frac{1}{d(d-1)} \sum_{i \neq j} \left| corr(X_i, X_j) \right| \tag{13}$$

where, $corr(X_i, X_j)$ is the Pearson correlation coefficient between the i-th and j-th dimension variables, given by:

$$corr(X_i, X_j) = \frac{\sum_{k=1}^{n} (x_{i,k} - \bar{x}_i)(x_{j,k} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{n} (x_{i,k} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^{n} (x_{j,k} - \bar{x}_j)^2}} \tag{14}$$

Here, n is the sample size, $\bar{x}_i$ and $\bar{x}_j$ are the sample means of the i-th and j-th dimensions, respectively.

Nonlinear index (η): This feature measures the degree of nonlinearity of the objective function, obtained by comparing the mean squared error (MSE) of linear and nonlinear models.

$$\eta = \frac{\mathrm{MSE}_{\mathrm{linear}} - \mathrm{MSE}_{\mathrm{nonlinear}}}{\mathrm{MSE}_{\mathrm{linear}}} \tag{15}$$

Here, MSE_{linear} is the MSE of a linear model on the samples, and $MSE_{nonlinear}$ is the MSE of a nonlinear model on the samples. Typically, a linear model and a nonlinear model are trained on the sample data, their MSE on the training samples is calculated, and then the nonlinear index η is computed. If the result is negative, it is set to 0.

Smoothness index (ξ): This feature measures the smoothness of the objective function in the input space, obtained by calculating the rate of change of the objective function value between sample points.

$$\xi = \frac{1}{1+\frac{1}{N_p}\sum_{i<j}\frac{\left|y_i - y_j\right|}{\left\|x_i - x_j\right\|}} \tag{16}$$

Here, $N_p = \binom{n}{2} = \frac{n(n-1)}{2}$ is the number of sample point pairs (*i.e.*, all pairwise combinations), y_i and y_j are the objective function values of sample points X_i and X_j respectively, $\left\|x_i - x_j\right\|$ is the Euclidean distance between sample points, *i.e.*:

$$\left\|x_i - x_j\right\| = \sqrt{\sum_{k=1}^{d}\left(x_{i,k} - x_{j,k}\right)^2} \tag{17}$$

Ruggedness index (r): The ruggedness index f8 is an indicator used to quantify the complexity or irregularity of the local structure of an optimization problem's objective function surface. It reflects how "sharp" or "chaotic" the function's variation is within a small region.

$$r_k = \frac{\mathrm{Var}\left(\lambda_1^{(k)}, \lambda_2^{(k)}, \cdots, \lambda_n^{(k)}\right)}{\mathrm{Mean}\left(\lambda_1^{(k)}, \lambda_2^{(k)}, \cdots, \lambda_n^{(k)}\right)} \tag{18}$$

Here $\lambda_i^{(k)}$ represents the local eigenvalues calculated for the k-th sampling point or within a certain sub-region, Var denotes variance, and Mean denotes the mean.

3.3. Basic Functions

Test functions have fitness landscapes with different characteristics, which also leads to inconsistent performance of a single model fitting different test functions. Analyzing the characteristics of basic functions and determining the most suitable surrogate models is the foundation for adaptively selecting surrogate models based on optimization problem features. The basic functions used in this paper are listed in Table 2, numbered F1 - F15 sequentially from the Zakharov function to the Griewank function.

Table 2. Basic function information table.

Functions	Function ID	Description	Variable Range	Dimensions	Optimal Value
Zakharov	F1	Function containing linear and quadratic terms	[−5, 10]	2/6/10	0

Continued

Rosenbrock	F2	Classic "banana" function with a narrow parabolic valley	[−2.048, 2.048]	2/6/10	0
Schaffer F7	F3	Highly multimodal function with numerous local optima	[−100, 100]	2/6/10	0
Rastrigin	F4	Cosine-modulated multimodal function with many local optima	[−5.12, 5.12]	2/6/10	0
Levy	F5	Multimodal function with oscillatory characteristics	[−10, 10]	2/6/10	0
Bent Cigar	F6	Elliptic paraboloid function with very high condition number	[−100, 100]	2/6/10	0
HGBat	F7	Hyper-ellipsoid bat function with multiple local optima	[−100, 100]	2/6/10	0
Katsuura	F8	Highly multimodal function with numerous local optima	[−100, 100]	2/6/10	0
Ackley	F9	Function with an almost flat region and many local optima	[−32, 32]	2/6/10	0
Modified Schwefel	F10	Modified version of the Schwefel function	[−100, 100]	2/6/10	0
Happy Cat	F11	Complex function with multiple local optima	[−100, 100]	2/6/10	0
Griewank-Rosenbrock	F12	Combined function of Griewank and Rosenbrock	[−100, 100]	2/6/10	0
Elliptic	F13	Axis-aligned quadratic function	[−100, 100]	2/6/10	0
Discus	F14	Elliptic paraboloid function with large condition number	[−100, 100]	2/6/10	0
Griewank	F15	Highly multimodal function with many regularly distributed local optima	[−600, 600]	2/6/10	0

3.4. Feature Extraction for Basic Functions

To determine the characteristic value of each basic function, statistical experiments were conducted. As the sample size increases, the function feature values tend to stabilize, and the value at this point is taken as the function's characteristic value. Research found that when the sample size reaches 200, the function value tends to stabilize. Statistics in Table 3 show the feature values of basic functions when the random sample size is 200.

3.5. Determination of Optimal Parameters

The experimental scheme for determining optimal parameters adopted a random search method to identify optimal parameters for multiple machine learning models. Random search randomly selects parameter combinations within a defined parameter grid and evaluates model performance via cross-validation, using the RandomizedSearchCV method from the sklearn.model_selection package. This method is more efficient than grid search because it does not evaluate all possible

parameter combinations but randomly selects combinations within a specified number of iterations, aiming to find good parameter configurations in a shorter time. Each model performed corresponding parameter search and best parameter selection for the 15 functions. The core parameter distribution refers to the most frequently occurring value for each individual parameter across all functions, and the optimal parameter combination refers to the most frequently occurring single set of parameters. This combination is used as a reference for subsequent experiments. Table 4 provides the optimal parameters for each model, which are used in all subsequent experiments.

Table 3. Feature values of basic functions with random sample size of 200.

Func	Mean Std	Dev	Skewness	Kurtosis	Correl	Nonlin	Smooth	Rugged
F1	3.43E+03	6.58E+03	2.65E+00	7.14E+00	5.00E−02	1.69E+00	0.00E+00	1.92E+00
F2	4.60E+02	5.88E+02	2.20E+00	4.98E+00	4.00E−02	1.71E+00	0.00E+00	1.27E+00
F3	1.79E+02	1.06E+02	6.70E−01	−5.00E−01	7.00E−02	1.72E+00	0.00E+00	5.90E−01
F4	3.75E+01	1.46E+01	0.00E+00	−2.00E−02	5.00E−02	1.77E+00	4.00E−02	3.80E−01
F5	1.73E+01	1.70E+01	1.67E+00	2.49E+00	5.00E−02	1.74E+00	3.00E−02	9.80E−01
F6	3.28E+09	2.88E+09	6.40E−01	−8.20E−01	6.00E−02	1.71E+00	2.45E−10	8.70E−01
F7	8.61E+03	5.26E+03	5.20E−01	−2.50E−01	5.00E−02	1.74E+00	0.00E+00	6.10E−01
F8	4.10E+01	2.21E+01	1.00E−02	−1.24E+00	4.00E−02	1.75E+00	3.00E−02	5.30E−01
F9	9.83E+00	2.51E+00	−8.40E−01	7.30E−01	5.00E−02	1.75E+00	2.10E−01	2.50E−01
F10	6.37E+02	3.36E+02	2.50E−01	−4.70E−01	3.00E−02	1.74E+00	0.00E+00	5.20E−01
F11	1.73E+03	1.05E+03	5.20E−01	−2.90E−01	4.00E−02	1.74E+00	0.00E+00	6.00E−01
F12	3.23E+15	5.56E+15	1.91E+00	2.77E+00	4.00E−02	1.72E+00	1.25E−16	1.72E+00
F13	3.28E+09	2.88E+09	6.40E−01	−8.20E−01	6.00E−02	1.71E+00	2.45E−10	8.70E−01
F14	3.61E+09	3.09E+09	4.90E−01	−1.17E+00	4.00E−02	1.72E+00	2.24E−10	8.50E−01
F15	6.31E+01	3.79E+01	5.20E−01	−2.50E−01	5.00E−02	1.74E+00	1.00E−02	6.00E−01

Table 4. Optimal parameter table for each model.

Model ID	Model	Optimal Parameters
M1	AdaBoostRegressor	‘estimator’: DecisionTreeRegressor(max_depth=5), ‘n_estimators’: 200, ‘loss’: ‘exponential’, ‘learning_rate’: 1.0, ‘random_state’: 42
M2	GBR	‘alpha’: 0.1, ‘learning_rate’: 0.05, ‘loss’: ‘huber’, ‘max_depth’: 2, ‘max_features’: None, ‘min_samples_leaf’: 1, ‘min_samples_split’: 5, ‘n_estimators’: 500, ‘random_state’: 42, ‘subsample’: 0.9

Continued

M3	GPR	alpha: 1e-05 kernel: 1**2 * RBF(length_scale=1) normalize_y: True optimizer: fmin_l_bfgs_b random_state: 42
M4	KernelRidge	alpha: 0.01, coef0: 1.0, degree: 3, gamma: 0.1, kernel: poly
M5	KNR	'algorithm': 'auto', 'leaf_size': 20, 'metric': 'euclidean', 'metric_params': None, 'n_jobs': -1, 'n_neighbors': 3, 'p': 4, 'weights': 'distance'
M6	LassoLars	'alpha': 10.0, 'fit_intercept': True, 'max_iter': 1000, 'normalize': True
M7	LGB	'colsample_bytree': 0.6, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'num_leaves': 15, 'reg_alpha': 0.1, 'reg_lambda': 0, 'subsample': 0.9
M8	LinearRegression	'fit_intercept': True, 'positive': True, 'tol': 0.001
M9	MLPRegressor	'hidden_layer_sizes': (50,), 'activation': 'relu', 'solver': 'adam', 'alpha': 0.001, 'learning_rate': 'constant', 'max_iter': 1000
M10	RFR	'n_estimators': 200, 'max_depth': None, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'log2'
M11	SGDRegressor	'alpha': 0.1, 'eta0': 1.0, 'learning_rate': 'constant', 'loss': 'epsilon_insensitive', 'max_iter': 1000, 'penalty': 'l1', 'shuffle': True

Continued

M12	SVR	'C': 100.0, 'epsilon': 0.1, 'gamma': 'scale', 'kernel': 'rbf'

Based on the determined optimal model parameters, statistical experiments were conducted to determine the optimal model for each function under different sample size conditions. Both training and test sets were generated using uniform random sampling, with the test set remaining consistent after generation. Table 5 lists the optimal models for basic functions as the sample size changes. The table shows that the optimal model stabilizes when the experimental sample size reaches 200.

Table 5. Optimal models for basic functions.

	F1	F2	F3	Z	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
10	M4	M10	M3		M4	M1	M4	M4	M3	M9	M2	M4	M2	M4	M4	M4
20	M3	M3	M9		M4	M2	M4	M4	M8	M9	M4	M4	M2	M4	M4	M4
30	M3	M3	M9		M4	M1	M4	M4	M7	M9	M3	M4	M2	M4	M4	M4
40	M3	M3	M9		M2	M2	M4	M4	M7	M9	M3	M3	M1	M4	M4	M4
50	M3	M3	M9		M2	M1	M4	M4	M11	M12	M3	M4	M1	M4	M4	M4
60	M3	M3	M9		M2	M1	M4	M3	M6	M12	M3	M4	M1	M4	M4	M4
70	M3	M3	M9		M2	M1	M4	M3	M6	M12	M3	M4	M1	M4	M4	M4
80	M3	M3	M4		M2	M2	M4	M3	M6	M12	M3	M4	M3	M4	M4	M4
90	M3	M3	M4		M2	M1	M4	M3	M6	M12	M3	M4	M3	M4	M4	M4
100	M3	M3	M9		M2	M1	M4	M3	M6	M12	M3	M4	M2	M4	M4	M4
110	M3	M3	M9		M2	M1	M4	M3	M6	M12	M3	M3	M2	M4	M4	M4
120	M3	M3	M9		M2	M1	M4	M4	M6	M12	M3	M4	M2	M4	M4	M4
130	M3	M3	M9		M2	M1	M4	M4	M8	M12	M3	M4	M2	M4	M4	M4
140	M3	M3	M9		M2	M2	M4	M4	M6	M12	M3	M4	M2	M4	M4	M4
150	M3	M3	M9		M2	M2	M4	M4	M6	M12	M3	M4	M2	M4	M4	M4
160	M3	M3	M9		M2	M2	M4	M4	M6	M12	M3	M4	M3	M4	M4	M4
170	M3	M3	M9		M2	M1	M4	M3	M6	M10	M3	M4	M2	M4	M4	M4
180	M3	M3	M4		M2	M2	M4	M3	M11	M10	M3	M4	M3	M4	M4	M4
190	M3	M3	M4		M2	M2	M4	M3	M6	M10	M3	M4	M3	M4	M4	M4
200	M3	M3	M9		M10	M2	M4	M3	M6	M10	M3	M4	M2	M4	M4	M4

3.6. SISSO-Based Model Selector

To ensure that similar models are encoded with closer numbers, this paper designs a model similarity metric. Models are trained with 200 training samples, and the R^2 values of each model fitting the 200 samples in the test set for each basic function are recorded, as shown in Table 6.

From Table 7, it is observed that for function F8, the corresponding test set R^2 values are all negative because it has many local optima, and all models fit poorly, making it unsuitable as a basic function. Therefore, this paper discards it. Based on the above table, the Euclidean distances between the 12 models are calculated to obtain the model correlation matrix in Table 5.

Table 6. Test set R^2 values under optimal parameters for different models.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
M1	0.95	0.92	0.31	0.68	0.96	1.00	0.92	−0.13	0.93	0.94	0.92	0.99	1.00	1.00	0.92
M2	0.94	0.91	0.22	0.70	0.97	1.00	0.98	−0.34	0.86	0.98	0.98	1.00	1.00	1.00	0.98
M3	1.00	1.00	0.12	0.00	0.90	1.00	1.00	−0.01	0.00	1.00	1.00	0.92	1.00	1.00	0.95
M4	0.99	0.96	0.37	0.51	0.60	1.00	1.00	−0.06	0.82	0.98	1.00	0.70	1.00	1.00	1.00
M5	0.97	0.93	0.15	0.35	0.71	0.96	0.96	−0.44	0.92	0.97	0.96	0.83	0.96	0.96	0.96
M6	0.48	0.17	0.00	0.00	0.00	0.00	−0.02	0.00	0.00	−0.01	−0.01	−0.06	0.00	−0.05	−0.01
M7	0.60	0.74	0.31	0.62	0.65	0.97	0.97	−0.19	0.90	0.98	0.97	0.80	0.97	0.98	0.97
M8	0.48	0.00	0.00	−0.02	0.00	0.00	−0.02	0.00	−0.01	−0.02	−0.02	−0.03	0.00	−0.04	−0.02
M9	−0.08	−0.21	0.37	0.46	0.53	−1.20	−0.70	−0.39	0.89	0.81	0.68	−0.26	−1.20	−1.32	0.86
M10	0.93	0.89	0.28	0.70	0.89	0.97	0.93	−0.26	0.94	0.94	0.93	0.93	0.97	0.96	0.93
M11	0.24	0.03	−0.05	−0.03	−0.06	−1.19	−0.12	−0.06	−0.02	−9.34	−25.87	−0.26	−1.19	−1.32	0.00
M12	0.22	0.69	0.34	0.49	0.59	−0.06	0.38	−0.18	0.93	0.99	0.88	−0.23	−0.06	−0.04	1.00

Table 7. Model correlation matrix.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
M1	0.00	0.15	1.18	0.52	0.47	3.29	0.54	3.33	4.66	0.11	29.08	2.41
M2	0.15	0.00	1.12	0.53	0.48	3.33	0.55	3.38	4.69	0.18	29.15	2.44
M3	1.18	1.12	0.00	1.06	1.01	3.13	1.24	3.18	4.82	1.20	29.15	2.64
M4	0.52	0.53	1.06	0.00	0.35	3.16	0.48	3.22	4.63	0.47	29.15	2.29
M5	0.47	0.48	1.01	0.35	0.00	3.11	0.52	3.16	4.57	0.43	29.09	2.28
M6	3.29	3.33	3.13	3.16	3.11	0.00	3.09	0.17	2.95	3.22	27.58	2.21
M7	0.54	0.55	1.24	0.48	0.52	3.09	0.00	3.13	4.47	0.47	29.10	2.16
M8	3.33	3.38	3.18	3.22	3.16	0.17	3.13	0.00	2.96	3.27	27.56	2.28
M9	4.66	4.69	4.82	4.63	4.57	2.95	4.47	2.96	0.00	4.59	28.48	2.52
M10	0.11	0.18	1.20	0.47	0.43	3.22	0.47	3.27	4.59	0.00	29.07	2.33
M11	29.08	29.15	29.15	29.15	29.09	27.58	29.10	27.56	28.48	29.07	0.00	28.81
M12	2.41	2.44	2.64	2.29	2.28	2.21	2.16	2.28	2.52	2.33	28.81	0.00

The Euclidean distances between models are treated as an adjacency matrix, where the graph nodes are model types and edges are Euclidean distances. The shortest path traversing all nodes in the graph is found. The longest edge on the path is selected, and numbering starts from one vertex of this edge. In this paper,

poorly performing models are discarded, and only the optimal models for the 200-sample point condition are used, *i.e.*, m3, m9, m10, m2, m4. Using brute-force solving for the Traveling Salesman Problem, the shortest path is found to be m3, m4, m2, m10, m9, which are then numbered sequentially as 1, 2, 3, 4, 5. The corresponding models are GPR (Gaussian Process Regression, M3), KernelRidge (Kernel Ridge Regression, M4), GBR (Gradient Boosting Regression, M2), RFR (Random Forest Regression, M10), and MLPRegressor (Multi-Layer Perceptron Regressor, M9). By rotating and translating the basic functions, a library of 165 transformed basic functions is obtained. These functions are sampled with a sample size of 200, and their features and optimal models are recorded, resulting in a table of feature and optimal model information for each basic function (see Table 8). Using the feature values as input and Best_ID as output, the SISSO algorithm is used to train a model, obtaining the model selector shown in Equation (19).

$$f(x) = \text{term1} + \text{term2} + \text{term3} + \text{constant} \tag{19}$$

where

Term1 = (−0.11E+01) * (abs((f7/f5) − abs(f6 − f8)) − (f7/f5))

Term2 = (0.64E−02) * ((f1 * f4)/((f4)^3 + (f5/f7)))

Term3 = (0.19E+00) * (((f3 − f4) − f8)/abs((f8)$^{-1}$ − f6))

Constant = (0.25E+01)

Table 8. Basic function characteristics and optimal model information.

Func	Sample Size	Mean Std	Dev	Skewness	Kurtosis	Correl	Nonlin	Smooth	Rugged	Best_ID
F1	200	3427.49	6584.49	2.65	7.14	0.05	1.69	0.00	1.92	M3
F2	200	460.13	588.09	2.20	4.98	0.04	1.71	0.00	1.27	M3
F3	200	178.78	105.63	0.67	−0.50	0.07	1.72	0.00	0.59	M9
F4	200	37.47	14.58	−0.00	−0.02	0.05	1.77	0.04	0.38	M10
F5	200	17.30	16.95	1.67	2.49	0.05	1.74	0.03	0.98	M2
F6	200	3.28E+09	2.88E+09	0.64	−0.82	0.06	1.71	2.45E−10	0.87	M4
F7	200	8609.39	5259.45	0.52	−0.25	0.05	1.74	0.00	0.61	M3
F8	200	41.01	22.07	0.01	−1.24	0.04	1.75	0.03	0.53	M6
F9	200	9.83	2.51	−0.84	0.73	0.05	1.75	0.21	0.25	M10
F10	200	637.43	336.19	0.25	−0.47	0.03	1.74	0.00	0.52	M3
F11	200	1730.55	1054.69	0.52	−0.29	0.04	1.74	0.00	0.60	M4
F12	200	3.23E+15	5.56E+15	1.91	2.77	0.04	1.72	1.25E−16	1.72	M2
F13	200	3.28E+09	2.88E+09	0.64	−0.82	0.06	1.71	2.45E−10	0.87	M4
F14	200	3.61E+09	3.09E+09	0.49	−1.17	0.04	1.72	2.24E−10	0.85	M4
F15	200	63.06	37.87	0.52	−0.25	0.05	1.74	0.01	0.60	M4

Mapping from optimization problem features to optimal model number is achieved by rounding the output of the model selector. The advantage of the

SISSO model selector is that it can discover complex nonlinear relationships between features and the optimal model with only a small amount of sample data. Finally, in the evaluation and validation phase, the system performs large-scale experiments on test functions, analyzes the stability of model selection under different sample sizes, determines the most frequently selected model at each sample size through multiple repeated experiments, and records the corresponding best performance metrics.

3.7. Adaptive Surrogate-Assisted Evolutionary Algorithm Based on SISSO Model Selector

This framework constructs an optimization problem model selection system based on an SISSO model selector [15]. The core idea is to statistically extract problem features during the optimization process, feed these features into the pre-trained SISSO model selector, and dynamically select the optimal surrogate model for different problems. The entire implementation follows a complete closed-loop process of "data collection—feature extraction—model selection—model-driven optimization".

First, in the data collection stage, intermediate solutions and their fitness values evaluated by the algorithm in each iteration are recorded to build a training sample database. Then, in the feature extraction stage, eight features of the optimization problem are statistically extracted based on the training sample database: including mean of objective values, standard deviation, skewness, kurtosis, correlation strength, nonlinear index, smoothness index, and ruggedness index. Next, in the model selection stage, these features are input into the pre-trained SISSO model selector. The framework flowchart is shown in Figure 2.

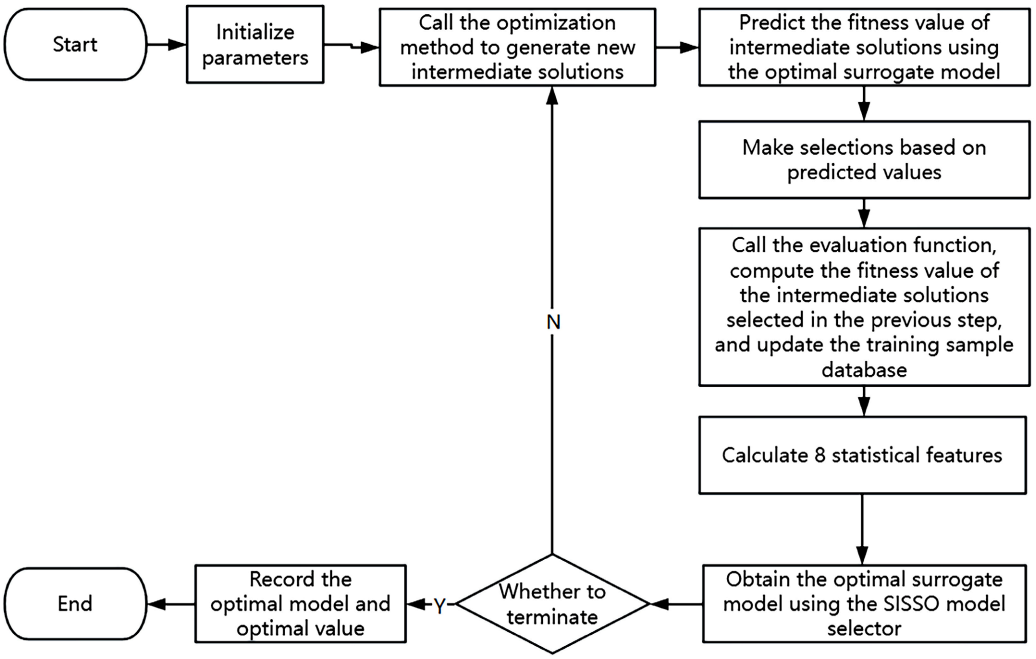

Figure 2. Adaptive surrogate-assisted evolutionary algorithm based on SISSO model selector.

3.7.1. Multi-Subpopulation Selection Strategy

To enhance the algorithm's global convergence capability and local exploitation efficiency on complex fitness landscapes, this paper proposes a multi-subpopulation selection strategy during the intermediate solution generation phase. Its core idea originates from the Multivariant Optimization Algorithm (MOA) [16]. In this strategy, intelligent search individuals are organized into multiple subpopulations with different responsibilities, dedicated to global exploration and local exploitation, respectively, thereby systematically coordinating the exploration-exploitation balance during the search process. This strategy divides the population into multiple subpopulations, allowing each subpopulation to independently perform sampling and evaluation, followed by explicit task allocation. Some subpopulations are assigned the role of global exploration, continuously searching the entire solution space to reduce the risk of getting trapped in local optima; the remaining subpopulations take on the role of local exploitation, performing fine-grained searches around the current best solution to improve solution quality within potential regions. This series of mechanisms makes it particularly suitable for complex optimization problems with multiple traps, deceptive, or randomly located optima, overall enhancing the probability of global convergence while also strengthening local exploitation capability.

3.7.2. Feature Extraction and Model Selection

After completing subpopulation division and sampling, the system enters the feature extraction and model selection stage. The core goal is to dynamically select the optimal surrogate model based on the characteristics of the current optimization problem, thereby achieving high-precision fitting of complex fitness landscapes. The feature extraction process obtains eight key features from the sampled data of each subpopulation, including objective value mean, standard deviation, skewness, and kurtosis that reflect data distribution characteristics, as well as correlation strength, nonlinear index, smoothness index, and ruggedness index that characterize the function structure. These features collectively describe the global and local properties of the problem. Based on the extracted features, the system employs the SISSO model selector as the classifier. This selector uses symbolic regression to screen out the most discriminative feature combinations from the high-dimensional feature space and constructs a mapping function as shown in Equation (19), outputting a model number corresponding to different surrogate models. During optimization, the system re-extracts features and inputs them into the SISSO model selector in each iteration, selecting the surrogate model most suitable for the current stage in real-time. This enables a flexible response to dynamic changes in the fitness landscape, improving fitting accuracy and optimization efficiency.

4. Experiments

To verify the reliability of the algorithm proposed in this paper, nine test functions proposed by CEC2022 are used to test the algorithm's performance. Among the

nine functions, they can be categorized into one unimodal function (f1), four basic functions (f2 - f5), and four composite functions (f9 - f12).

4.1. Parameter Settings

The four models—ElasticNet, Lars, Lasso, and Ridge—used as baselines in this paper are all independent optimization algorithms.

During the experimental process, the following parameter settings were found to achieve the best results: problem dimension D = 2, maximum iterations I = 7, population size P = 6, initial sampling points S = 20, subpopulation configuration of 4 subpopulations with 100 samples each, Gaussian sampling, radius scaling 0.7, each function run 30 times. Comparisons were made with four popular surrogate models in recent years: ElasticNet, Lars, Lasso, and Ridge. The symbols "+", "−", and "≈" indicate that the proposed SISSO-ASEA algorithm is superior, inferior, or shows no significant difference compared to other algorithms, respectively.

4.2. Overall Performance Comparison

To verify the performance of the SISSO-ASEA algorithm, we integrated the SISSO optimization selector framework, named SISSO-Adaptive-Optimizer (SAO), which incorporates the complete closed loop of "sampling → feature extraction → model selection → optimization search". This paper selected four models—ElasticNet, Lars, Lasso, and Ridge—for comparison with SAO. Table 9 shows the mean and variance of the error obtained from 30 independent runs of SAO and the other four models in 2 dimensions. The results show that SAO exhibits significant advantages on the vast majority of test functions. Among the nine test functions, SAO outperforms ElasticNet on 7 functions, Lars on 2 functions, Lasso on 7 functions, and Ridge on 7 functions, demonstrating its good generalization ability and stability. Particularly on function f9, SAO's mean error is far lower than that of the comparison algorithms, indicating its outstanding capability in handling highly nonlinear, multimodal problems. Furthermore, on functions f2 and f4, SAO consistently achieves lower mean errors, showing its strong adaptability and predictive accuracy for different function characteristics. Although on individual functions like f1 and f3, SAO performs comparably to some algorithms, its overall performance is significantly leading, with relatively stable variance control. This further verifies that SAO, as an adaptive surrogate model optimization framework based on the SISSO algorithm, has strong competitiveness and practicality in expensive optimization problems.

Table 9. Comparison results of SAO and 4 models on CEC2022 Test functions in 2 dimensions.

Function	Error	SAO	ElasticNet	Lars	Lasso	Ridge
F1	Mean	1.30E+02	1.05E+03	8.23E+01	1.08E+03	1.07E+03
	Variance	7.22E+05	8.91E+05	6.18E+06	8.92E+05	8.02E+05
F2	Mean	2.00E−02	5.40E−01	9.00E−02	6.90E−01	5.10E−01
	Variance	3.71E−03	1.58E+00	1.25E+00	6.96E+00	1.41E+00

Continued

F3	Mean	1.00E−02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Variance	8.80E−05	5.40E−05	6.60E−05	6.90E−05	1.51E−04
F4	Mean	2.96E+00	3.74E+00	2.98E+00	3.59E+00	3.48E+00
	Variance	4.54E+01	1.74E+01	3.44E+01	3.26E+01	1.80E+01
F5	Mean	6.00E−02	1.90E−01	1.00E−02	1.70E−01	1.60E−01
	Variance	2.00E−02	2.28E−02	3.18E−02	1.36E−02	6.03E−02
F9	Mean	1.03E+00	5.00E+01	2.04E+01	4.06E+01	5.95E+01
	Variance	2.97E−01	8.11E+03	3.87E+04	4.76E+04	3.00E+04
F10	Mean	1.19E+02	1.26E+02	1.10E+02	1.28E+02	1.27E+02
	Variance	1.50E+03	9.89E+02	2.50E+02	1.18E+03	8.23E+02
F11	Mean	1.86E+02	2.03E+02	1.05E+02	2.03E+02	2.04E+02
	Variance	1.13E+04	9.03E+03	1.08E+04	1.07E+04	1.10E+04
F12	Mean	3.21E+01	8.99E+01	1.85E+01	8.58E+01	8.49E+01
	Variance	4.67E+03	9.86E+03	8.86E+03	9.56E+03	5.84E+03
		+	7	2	7	7
		−	1	6	1	1
		≈	1	1	1	1

4.3. Convergence Analysis

To more intuitively demonstrate the performance of SISSO-ASEA, we compared the convergence of SISSO-ASEA and the other four algorithms in 2 dimensions. The experiment selected f1, f5, f9, and f12 from the CEC2022 test functions. From Figure 3, it can be seen that SISSO-ASEA, compared to the other four surrogate models in later stages, can still search for intermediate solutions across different test functions, exhibiting significant continuous convergence capability and better final solution quality.

4.4. Comparative Experiments

To verify the algorithm's performance on ten-dimensional optimization problems, this paper conducted comparative experiments. The comparison results on CEC2022 test functions in ten dimensions are shown in Table 7. Through comparative analysis of performance on two-dimensional and ten-dimensional optimization problems, we found that the SAO algorithm exhibits significant advantages on ten-dimensional problems. Statistical data from Table 10 shows that SAO performs better than ElasticNet, Lasso, and Ridge on 8 functions each, showing inferiority to Lars on only 1 function. This indicates that as problem dimensionality increases, the SAO algorithm can better handle complex optimization problems and maintain high solution accuracy. In contrast, SAO's performance on two-dimensional problems shows differentiated characteristics. For the three models, ElasticNet, Lasso, and Ridge, SAO performs better on 7 functions; however, it is better than Lars

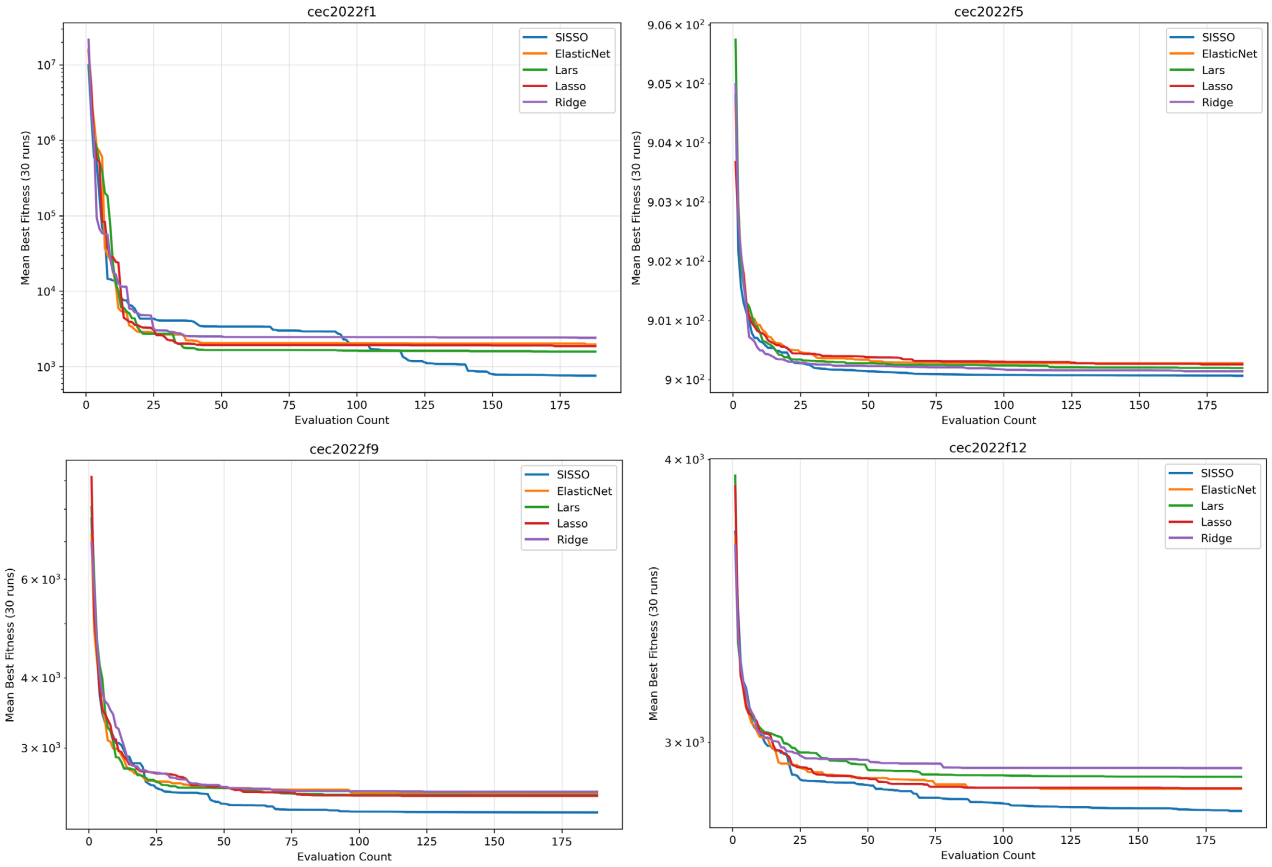

Figure 3. Convergence plots of different algorithms.

Table 10. Comparison results of SAO and 4 models on CEC2022 test functions in 10 dimensions.

Function	Error	SAO	ElasticNet	Lars	Lasso	Ridge
F1	Mean	1.45e+04	3.18e+04	2.38e+04	3.35e+04	3.29e+04
	Variance	4.36e+03	9.93e+03	6.94e+03	1.13e+04	1.05e+04
F2	Mean	1.14E+02	3.95E+02	5.67E+02	4.40E+02	4.43E+02
	Variance	3.32E+01	2.07E+02	2.65E+02	2.15E+02	2.08E+02
F3	Mean	1.10E−01	4.10E−01	3.80E−01	4.60E−01	4.50E−01
	Variance	9.00E−02	1.30E−01	1.20E−01	1.30E−01	1.30E−01
F4	Mean	1.16E+02	1.73E+02	1.74E+02	1.84E+02	1.83E+02
	Variance	3.00E+01	2.94E+01	3.64E+01	2.50E+01	2.30E+01
F5	Mean	1.17E+00	6.87E+00	5.49E+00	7.01E+00	7.19E+00
	Variance	6.80E−01	1.84E+00	1.80E+00	2.24E+00	2.19E+00
F9	Mean	4.39E+02	7.32E+02	7.47E+02	7.54E+02	7.11E+02
	Variance	1.13E+02	2.79E+02	2.19E+02	2.75E+02	2.65E+02
F10	Mean	4.92E+02	1.06e+03	1.17e+03	1.31e+03	1.34e+03
	Variance	1.98E+02	8.52E+02	8.21E+02	8.91E+02	8.77E+02

Continued

F11	Mean	1.74E+02	1.05e+03	7.09E+02	1.01e+03	1.04e+03
	Variance	2.42E+02	2.59E+02	3.95E+02	2.83E+02	3.23E+02
F12	Mean	2.22E+02	2.03E+02	2.50E+02	1.93E+02	1.99E+02
	Variance	3.04E+01	3.70E+01	4.07E+01	2.15E+01	3.47E+01
		+	8	8	8	8
		−	0	0	1	1
		≈	1	1	0	0

model on only 2 functions, forming a sharp contrast with the ten-dimensional case. It is particularly noteworthy that on function F3, SAO performs poorly against all compared models in the two-dimensional problem, while in the ten-dimensional problem, it shows only slight inferiority to some models. From the perspective of algorithm robustness, SAO's variance is generally lower in ten-dimensional problems, indicating better stability. This dimensional adaptability suggests that the SAO algorithm possesses stronger search capability and convergence characteristics when handling high-dimensional optimization problems.

5. Conclusion

Aiming at the limitation that a single surrogate model struggles to adapt to complex dynamic optimization processes in surrogate-assisted evolutionary algorithms, this paper proposes an adaptive surrogate model selection framework based on an SISSO model selector (SISSO-ASEA). This method dynamically extracts an eight-dimensional feature vector from the optimization problem and, through a pre-trained SISSO symbolic regression model, maps the problem features to the optimal surrogate model type, achieving a closed-loop adaptive process. Experiments prove the high efficiency and effectiveness of the proposed method.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Shao, W.W, Zhu, Y.G. and Hu, C. (2024) Application of Improved RRT Algorithm in Path Planning. *Journal of Nanyang Normal University*, **23**, 58-63.

[2] Yang, Y., Liu, J. and Tan, S. (2021) A Multi-Objective Evolutionary Algorithm for Steady-State Constrained Multi-Objective Optimization Problems. *Applied Soft Computing*, **101**, Article ID: 107042. https://doi.org/10.1016/j.asoc.2020.107042

[3] Qiu, X., Xu, J., Xu, Y. and Tan, K.C. (2018) A New Differential Evolution Algorithm for Minimax Optimization in Robust Design. *IEEE Transactions on Cybernetics*, **48**, 1355-1368. https://doi.org/10.1109/tcyb.2017.2692963

[4] Wei, J. and Niu, H. (2022) Offline Data-Driven Particle Swarm Optimization Assisted by Selective Surrogate Ensembles and Hybrid Search Strategies. 2022 34*th Chinese Control and Decision Conference* (*CCDC*), Hefei, 15-17 August 2022, 4010-4016.

https://doi.org/10.1109/ccdc55256.2022.10033635

[5] Su, G., Sun, W. and Zhao, Y. (2023) Machine Learning Classifier-Based Dynamic Surrogate Model for Structural Reliability Analysis. *Structure and Infrastructure Engineering*, **21**, 525-537. https://doi.org/10.1080/15732479.2023.2218847

[6] Luo, S. (2023) Synthetic Minority Oversampling Technique Based on Adaptive Noise Optimization and Fast Search for Local Sets for Random Forest. *International Journal of Pattern Recognition and Artificial Intelligence*, **37**, Article ID: 2259038. https://doi.org/10.1142/s0218001422590388

[7] Li, M., Sadoughi, M., Hu, Z. and Hu, C. (2020) A Hybrid Gaussian Process Model for System Reliability Analysis. *Reliability Engineering & System Safety*, **197**, Article ID: 106816. https://doi.org/10.1016/j.ress.2020.106816

[8] He, C., Zhang, Y., Gong, D. and Ji, X. (2023) A Review of Surrogate-Assisted Evolutionary Algorithms for Expensive Optimization Problems. *Expert Systems with Applications*, **217**, Article ID: 119495. https://doi.org/10.1016/j.eswa.2022.119495

[9] Zuo, S.Y. (2024) Parameter Analysis Modeling and Application Based on Additive Gaussian Process. Master's Thesis, Dalian University of Technology.

[10] Zhang, M. (2025) Electric Vehicle Charging Load Prediction Based on Optimized Random Forest Algorithm. *Digital Communication World*, No. 6, 26-28.

[11] Liang, X. (2024) Prediction of Landslide Displacement Data Based on Support Vector Regression. Master's Thesis, Lanzhou University of Technology.

[12] Xiao, T.C. (2022) Optimal Latin Hypercube Design under Symmetric Variable Conditions. Master's Thesis, Central University of Finance and Economics.

[13] Gong, L.L. (2022) Research on Software Defect Prediction Method Combined with Multiple Features. Master's Thesis, Hangzhou University of Electronic Science and Technology.

[14] Cui, J.S., Liu, X.C.,Yang, M.H., *et al.* (2017) Automatic Selection Framework and Empirical Analysis of Optimization Algorithm Based on Meta-Learning Recommendation. *Computer Applications*, **37**, 1105-1110.

[15] Li, B., Wang, D., Yu, M., He, C., Li, X., Zhai, J., *et al.* (2025) Accelerating Multi-Objective Catalytic Material Design: A Model-Based Method. *Chinese Chemical Letters*, **36**, Article ID: 110454. https://doi.org/10.1016/j.cclet.2024.110454

[16] Li, B., Chen, J., Shi, X., Zhang, Y., Lv, D., Liu, L., *et al.* (2016) On the Convergence of Multivariant Optimization Algorithm. *Applied Soft Computing*, **48**, 230-239. https://doi.org/10.1016/j.asoc.2016.07.001