A Surrogate-Assisted Evolutionary Algorithm for Space Component Thermal Layout Optimization

Han, Lei; Wang, Handing; Wang, Shuo

DOI:
10.34133/2022/9856362

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Han, L, Wang, H & Wang, S 2022, 'A Surrogate-Assisted Evolutionary Algorithm for Space Component Thermal Layout Optimization', *Space: Science and Technology (United States)*, vol. 2022, 9856362. https://doi.org/10.34133/2022/9856362

AAAS
Space: Science & Technology
Volume 2022, Article ID 9856362, 15 pages
https://doi.org/10.34133/2022/9856362

Space: Science & Technology
A SCIENCE PARTNER JOURNAL

Research Article

A Surrogate-Assisted Evolutionary Algorithm for Space Component Thermal Layout Optimization

Lei Han,[1] Handing Wang ,[1] and Shuo Wang[2]

[1]*School of Artificial Intelligence, Xidian University, Xi'an, China*
[2]*School of Computer Science, The University of Birmingham, Birmingham, UK*

Correspondence should be addressed to Handing Wang; hdwang@xidian.edu.cn

Received 23 March 2022; Accepted 17 July 2022; Published 30 July 2022

In space engineering, the electronic component layout has a very important impact on the centroid stability and heat dissipation of devices. However, the expensive thermodynamic simulations in the component thermal layout optimization problems bring great challenges to the current optimization algorithms. To reduce the cost, a surrogate-assisted evolutionary algorithm with restart strategy is proposed in this work. The algorithm is consisted of the local search and global search. A restart strategy is designed to make the local search jump out of the local optimum promptly and speed up the population convergence. The proposed algorithm is compared with two state-of-the-art algorithms on the CEC2006, CEC2010, and CEC2017 benchmark problems. The experiment results show that the proposed algorithm has a high convergence speed and excellent ability to find the optimum in the expensive constrained optimization problems under the very limited computation budget. The proposed algorithm is also applied to solve an electronic component layout optimization problem. The final results demonstrate the good performance of the proposed algorithm, which is of great significance to the component layout optimization.

1. Introduction

In space engineering, the layout of electronic components on circuit board has an important impact on the working state of components. However, the diversity of components, a variety of spatial location constraints, and thermal performance constraints have brought great challenges to the layout design. Moreover, the thermodynamic simulation is very expensive and time-consuming. For convenience, the component layout optimization problems (CLOPs) can be formulated as the expensive constrained optimization problems (ECOPs) as follows:

$$\begin{aligned} &\text{minimize } f(\mathbf{x}), \\ &\text{subject to } g_j(\mathbf{x}) \le 0, j = 1, \cdots, m, \end{aligned} \tag{1}$$

where $\mathbf{x} = [x_1, \cdots, x_d]$ is the decision vector, $x_k \in [l_k, u_k], k = 1, \cdots, d$. When the function evaluations $f(\cdot)$ and $g(\cdot)$ are time-consuming or expensive experiments [1, 2] which generally cannot be described by the analytic formula, they are expensive optimization problems. For example, the thermal layout simulation of electronic components in $f(\cdot)$ and $g(\cdot)$ is time-consuming, so it is expensive to evaluate the performance of a solution. Therefore, it is unrealistic to use a mount of real function evaluations in the optimization process. The ECOPs should be solved under the limited function evaluations. In addition, the ECOPs need the specific constraint handling strategies.

Evolutionary algorithms (EAs) have outstanding advantages in solving black-box optimization problems [3]. EAs only rely on the input and output to keep approaching the optimum through its iterations, which do not need the mathematical formulation of the problems. EAs are the biological heuristic algorithms which cannot guarantee to get the global optimum but an approximated optimum. The representative EAs are the genetic algorithm (GA) [4] and the differential evolution (DE) [5].

Although EAs can deal with the black-box problems, they need a lot of evaluations. The high cost still hinders solving the problems. Thus, the surrogates are used for modeling the evaluation process to replace some real evaluations with the model prediction in the optimization

process, which is called the surrogate-assisted evolutionary optimization [6–8]. Many machine learning techniques have been applied to construct the surrogate models, such as the Gaussian process regression (GP) [9], the radial basis function network (RBFN) [10], the polynomial regression (PR) [11], the support vector machine (SVM) [12], and the artificial neural network (ANN) [13]. For finding the optimum and improving the surrogates, there have many strategies which are generally called the model management strategies designed for selecting the candidate solutions for real evaluations [14, 15].

Another core issue of ECOPs is how to handle constraints. There are mainly three kinds of constraint handling methods. The first is the penalty function method [16, 17] to impose penalties on the constraints. The second is the feasibility rule proposed in [18]. The feasible solutions and infeasible solutions are compared by the specific rules. The stochastic ranking proposed in [19] is also a kind of feasibility rule method. The third is the multiobjective method [20, 21] which regards the constraints as the objective to transform the origin constrained optimization problems into the multiobjective optimization problems.

There are some representative algorithms proposed to address the ECOPs. For example, a global and local surrogate-assisted differential evolution algorithm (GLoSADE) is proposed in [22]. In its global search stage, the algorithm adopts two DE variants for prescreening search and the generalized regression neural network for modeling the objective functions and the constraint functions. In its local search stage, the interior point method is applied to find the best solution in the local area and the RBFN models are established for the objectives and the constraints. The constraint handling adapts the feasibility rule for comparing the solutions. An evolutionary algorithm with multiple penalty functions and multiple local surrogates (MPMLS) is proposed in [23]. The algorithm decomposes the search space into multiple subregions and builds one local surrogate model for each subregion. The penalty function method is applied to deal with the constraints.

In this work, we aim to design a surrogate-assisted evolutionary algorithm for solving the ECOPs efficiently. The local search can converge rapidly but easily fall into local optimum while the global search is on the contrary. Thus, we propose a restart strategy for combining the advantages of both strategies. Meanwhile, a local search method is proposed for enhancing local exploitation, and a new constraint violation calculation method is developed for handling the constraints. In addition, the modeling efforts are reduced by the proposed model management strategy. We term the proposed algorithm surrogate-assisted evolutionary algorithm with restart strategy (SAEA-RS).

The rest of this paper is organized as follows. Section 2 gives some preliminaries to establish a basis for the following discussion. Section 3 introduces the framework and some main parts in the proposed algorithm. Section 4 presents ablation experiments and comparative experiments with other algorithms. Section 5 presents the experimental results of the proposed algorithm on a circuit component thermal layout optimization problem. Section 6 concludes the paper and discusses the future research directions for the expensive constrained optimization.

2. Preliminaries

2.1. Radial Basis Function Network. RBFN is a three-layer forward network which are the input layer, the hidden layer, and the output layer. It has excellent fitting ability, simple structure, and high training speed. Thus, it has been widely used as the surrogate model in surrogate-assisted evolutionary optimization. Given that $\{(\mathbf{x}_i, y_i)|i = 1, \cdots, N\}$, RBFN can be described in the function expression as follows:

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

where w_i is the weight and $\phi(.)$ is the basis function. The w_i can be calculated using the following expression:

$$\begin{aligned} \mathbf{w} &= \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathbf{y}, \\ \Phi &= \begin{bmatrix} \phi(\mathbf{x}_1 - \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_1 - \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N - \mathbf{x}_1) & \cdots & \phi(\mathbf{x}_N - \mathbf{x}_N) \end{bmatrix}, \end{aligned} \tag{3}$$

where $\mathbf{w} = [w_1, \cdots, w_N]^T$ and $\mathbf{y} = [y_1, \cdots, y_N]^T$. The uncertainty of the solution can well represent the confidence of the current solution, which can guide the generation and retention of the solutions and further improve the accuracy of the model. The uncertainty can be calculated through the formula below:

$$\sigma^2(\mathbf{x}) = -\Phi(\mathbf{x})\Phi^{-1}\Phi(\mathbf{x})^T, \tag{4}$$

where $\Phi(\mathbf{x}) = [\phi(\mathbf{x} - \mathbf{x}_1), \cdots, \phi(\mathbf{x} - \mathbf{x}_N)]$.

2.2. Differential Evolution. DE is a classical EA which has a fast convergent speed. Meanwhile, it has very excellent robustness. Also, it is inherently parallelized which is very convenient for parallel computing. Mutation, crossover, and selection are three main operators of DE.

The mutation operator has many variants with different characteristics. The most representative variant is described below:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \times (\mathbf{x}_{r2} - \mathbf{x}_{r3}), k = 1, \cdots, N. \tag{5}$$

For each $\mathbf{v}_i$, the $r1$, $r2$, and $r3$ are three different integers randomly selected from $\{1, 2, \cdots, i-1, i+1, \cdots, N\}$. Thus, $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, and $\mathbf{x}_{r3}$ are the individuals in the population. F is the scaling factor, and N is the population size.

The crossover operator can be described as follows. The $\mathbf{u}_{i,k}$ is the combination of $\mathbf{v}_{i,k}$ and $\mathbf{x}_{i,k}$ through the crossover. The CR is the crossover probability, and j_{rand} is a random

location. If a random number between 0 and 1 is less than or equal to CR or $j = j_{\text{rand}}$, the $\mathbf{u}_{i,k}$ should be the $\mathbf{v}_{i,k}$. Otherwise, the $\mathbf{u}_{i,k}$ should be the $\mathbf{x}_{i,k}$.

$$\mathbf{u}_{i,k} = \begin{cases} \mathbf{v}_{i,k}, & \text{if rand } (0,1) \le \text{CR or } j = j_{\text{rand}}, \\ \mathbf{x}_{i,k}, & \text{otherwise.} \end{cases} \tag{6}$$

The selection operator has many approaches, such as the roulette wheel selection [24], the tournament selection [25], feasibility ranking, and stochastic ranking. They have different selection intensity that lead to different convergent speed.

2.3. Constraint Handling. For constrained optimization problems, the core issue is how to handle their constraints. The common approaches can be concluded in three categories.

The first method is the penalty function. Through the penalty function, the objective and the constraint function are integrated into a fitness function, so the constrained problems are transformed into an unconstrained problems. A prominent problem in this approach is how to set the proper penalty coefficients, which greatly affects the optimization performance.

The second method is the feasibility rule, which provides a comparison rule among the feasible solutions and the infeasible solutions. The feasibility rule has three basic rules.

(i) Between two feasible solutions, the solution with a better objective value is the better one

(ii) Between a feasible solution and a infeasible solution, the feasible solution is the better one

(iii) Between two infeasible solutions, the solution with a less constraint violation value is the better one

Based on the feasibility rule, there are many similar rules proposed, such as the stochastic ranking, ε-constrained method [26], and diversity maintenance [27].

The third method is the multiobjective method. The constraints or constraint violation are considered as the objectives. Therefore, the constrained optimization problems are transformed into the multiobjective optimization problems. Hence, plenty of multiobjective optimization algorithms are available for solving the constrained optimization problems.

3. Proposed Algorithm

3.1. Framework. The flowchart of the proposed algorithm is shown in Figure 1. The algorithm has two coordinate optimization processes. One is a global search process, and another one is a local search process. The local search can converge rapidly but easily fall into the local optimum. In order to solve this problem, a restart strategy is introduced to restart the population when the search falls into the local optimum or falls behind the global search. A local search method is proposed to exploit the local region efficiently. The global search adopts the DE as the optimizer. To reduce the number of the real function evaluations, the RBFN models are established for each real objective and constraint functions to evaluate the newly generated individuals. The RBFN models are updated during the optimization process by the model management strategy proposed in this paper. For adapting to the different characteristics of the two optimization processes, the feasibility ranking is applied to the local search, and the stochastic ranking is applied to the global search. The feasibility ranking focuses on the convergence which is more suitable for the local search, while the stochastic ranking does not only retain the more feasible solutions but also preserves some infeasible solutions closed to the feasible region to maintain the population diversity. So, the stochastic ranking is adopted in the global search.

3.2. Proposed Constraint Violation. In the feasibility rule, the degree of constraint violation is generally defined as the sum of all constraint violations, which can be described as follows:

$$CV(\mathbf{x}) = \sum_{j=1}^{m} \max\left(0, g_j(\mathbf{x})\right). \tag{7}$$

We proposed a new constraint violation calculation method as below:

$$CV_{\text{new}}(\mathbf{x}) = \max\left(\max\left(0, g_1(\mathbf{x})\right), \cdots, \max\left(0, g_m(\mathbf{x})\right)\right), \tag{8}$$

where $CV_{\text{new}}(\mathbf{x})$ chooses the maximum of all the constraint violations as the constraint violation of $\mathbf{x}$. For two infeasible solutions, the one closer to the feasible region should be the better one. The infeasible solution closer to the feasible region is the one with smaller maximum of all the constraint violations, which is similar to the "cask effect." However, an infeasible solution with a smaller sum of constraint violation may not be the one closer to the feasible region. The reason is that an infeasible solution with a smaller sum of all constraint violations may have a large maximum of all constraint violations. Therefore, the maximum of all constraint violations can better compare the infeasible solutions.

3.3. Restart Strategy. The local search has a fast convergence ability, but it also easily falls into the local optimum. Therefore, a restart strategy is proposed in order to make the local search jump out of the current local optimum. The pseudocode of the restart strategy is shown in Algorithm 1. Firstly, the restart condition of local search population is defined as that the historical optimal solution obtained by local search is worse than that obtained by global search, or the local population diversity is lower than the diversity threshold. The diversity threshold is calculated by the following formula:

$$\text{PD}_{\text{threshold}} = \sum_{k=1}^{d} \delta(u_k - l_k), \tag{9}$$

where δ is a threshold factor. It should be between 0 and 1. The smaller it is, the higher the tolerance of losing

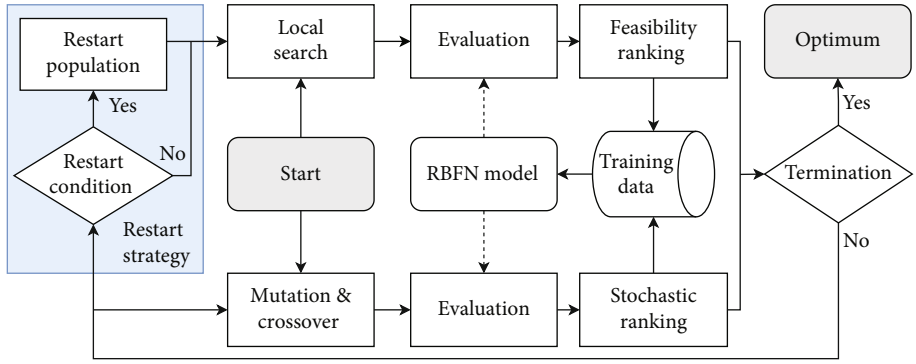

Figure 1: A diagram of the proposed algorithm.

Input: P_{local}: the local search population, l_{best}: the historical best individual found by local search, P_{global}: the global search population, g_{best}: the historical best individual found by global search, a_{best}: the historical best individual found by global search and local search, $\text{PD}_{\text{threshold}}$: the threshold of the population diversity, PD_{local}: the population diversity of the P_{local}, P_{new}: the new population generated by the restart strategy, P_{search}: the population generated by the local search
Output: P_{local}: the local population after the restart strategy
Calculate $\text{PD}_{\text{threshold}}$ and PD_{local}
1 Calculate $\text{PD}_{\text{threshold}}$ by the Equation (9).
2 Calculate PD_{local} by the Equation (10).
//Restart population
3 **if** P_{local} *is not in the protection period(within 5 iterations after restarting the population)* **then**
4 **if** g_{best} is better than l_{best} or $\text{PD}_{\text{local}} < \text{PD}_{\text{threshold}}$ **then**
5 Generate the P_{new} by the formulae in Equation (11) (*The x_{r1} in Equation ((11)) is always a_{best} here, and the x_{r2} and the x_{r3} are two different individuals randomly selected from the P_{global}.*).
6 Set P_{local} as P_{new}.
7 **end**
8 **end**

Algorithm 1: Pseudocode of restart strategy.

population diversity is. The local population diversity is calculated by the formulae below:

$$\text{PD}_{\text{local}} = \sum_{i=1}^{N}\sum_{k=1}^{d}\left|x_{i,k} - \bar{x}_k\right|, \quad \bar{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{i,k}, k = 1, \cdots, d. \tag{10}$$

It should be noted that the local population has a protection period which has been defined within 5 iterations after each restart. Therefore, the local population will not be restarted during the protection period. The restart population is generated by Equation (11), but x_{r1} is always the historical best individual in this part. Then, the local population will be set to the new restarted population.

3.4. Local Search. The global search adopts the mutation and crossover operator in DE to generate the new individuals of global population. Then, the new population and parent population are combined and evaluated by the RBFN models. Finally, the stochastic ranking is used to select N individuals as the offspring population. In order to enhance the local exploitation, we propose a new local search method. After the restart strategy, the local population uses Equation (11) to generate new individuals, and the newly obtained population are combined with the parent population. The combined population are sorted by the feasibility rule, and the best N individuals are selected as the offspring population.

$$\begin{aligned} \mathbf{v}_{i,k} &= \mathbf{x}_{r1,k} + X_k, \\ X_k &\sim N(0, \sigma_k), \\ \sigma_k &= abs(\mathbf{x}_{r2,k} - \mathbf{x}_{r3,k}), \\ k &= 1, \cdots, d, \end{aligned} \tag{11}$$

where $r1$, $r2$, and $r3$ are three different integers randomly sampled from $\{1, 2, \cdots, i-1, i+1, \cdots, N\}$ and X is a random variable with a mean value of 0 and a variance of σ. The operator is very similar to the mutation operator of DE, but it can generate more diverse local solutions for enhancing the local exploitation ability.

Input: P_{local}: the local search population, P_{global}: the global search population, *iter*: the current iterations, d_{interval}: the number of interval iterations, i_b: the selected individual from the P_{local}, i_u: the selected individual from the P_{global}, D_{train}: the training set of the surrogate models
Output: $M_{\text{objective}}$: the objective model, $M_{\text{constraint}}$: the constraint models
if mod (iter, d_{interval}) == 0 **then**
  P_{local} is sorted according to the feasibility rule from the best to the worst.
  P_{global} is sorted according to the order of uncertainty from the maximum to the minimum.
  Let $i_b = P_{\text{local}}(1)$.
  if i_b is in the D_{train} **then**
    Select $P_{\text{local}}(2)$ to $P_{\text{local}}(|P_{\text{local}}|)$ as i_b in order until i_b is not in the D_{train}.
    if i_b is still in the D_{train} **then**
      Generate an individual randomly in the search space as the i_b.
    end
  end
  Let $i_u = P_{\text{global}}(1)$.
  if i_u is in the D_{train} **then**
    Select $P_{\text{global}}(2)$ to $P_{\text{global}}(|P_{\text{global}}|)$ as i_u in order until i_u is not in the D_{train}.
    if i_u is still in the D_{train} **then**
      Generate an individual randomly in the search space as the i_u.
    end
  end
  Evaluate the i_b and the i_u with the real functions.
    $D_{\text{train}} = D_{\text{train}} \cup i_b \cup i_u$.
    Train the $M_{\text{objective}}$ and the $M_{\text{constraint}}$ with the D_{train}.
end

Algorithm 2: Pseudocode of model management.

3.5. Model Management. For improving the models' prediction ability, a model management strategy is proposed as shown in Algorithm 2. The sampling frequency is set to once every d_{interval} generations. Therefore, the models $M_{\text{objective}}$ and $M_{\text{constraint}}$ will be updated every d_{interval} generations. The prediction ability of the model can be divided into the local prediction ability and the global prediction ability. In order to improve the local prediction ability, the best individual i_b which has not been evaluated by the real function in the local search population P_{local} is selected as the sampling point. In order to improve the global prediction accuracy, the most uncertain individual i_u calculated by Equation (4) in the global search population P_{global} is selected as the sampling point. Similar to the local search, if the individual has been evaluated by the real function, the second uncertain individual is selected, and so on. In the above two sampling methods, if all individuals of the population have been evaluated by the real function, a random individual in the search space is chosen as the sampling point. The sampling points are evaluated by real function and added to the training dataset D_{train} for training the surrogate models $M_{\text{objective}}$ and $M_{\text{constraint}}$ again.

4. Comparative Experiment

4.1. Ablation Experiment. In order to verify the performance of each part of the proposed algorithm, the ablation experiments are carried out on the restart strategy and the local search. Therefore, there are three different algorithms which are a basic SAEA without the restart strategy and the local search and a SAEA with the local search(SAEA-LS) and the SAEA-RS. In this experiment, the surrogate model of all algorithms adopts the RBFN model, and the kernel function adopts cubic kernel function. The model management strategy adopts the method in Algorithm 2, where the sampling frequency d_{interval} is set to 5. Since the initial sampling takes 100 real function evaluations, there remain 200 real function evaluations that can be used during the iterations. Since the RBFN models are updated every 5 generations and each update costs 2 real function evaluations, the number of evolutionary generations is set to 500. The constraint violation calculation method is the proposed CV_{new}.

Firstly, the basic algorithm is the SAEA without the restart strategy and the local search. The initial population whose size is 100 is generated by the Latin hypercube sampling method [28]. The main optimizer is the DE whose scaling factor F is set to 0.8, and the crossover probability CR is set to 0.4. The selection adopts the stochastic ranking whose probability P_f is set to 0.45. The model management adopts the method proposed in the Algorithm 2. It should be noted that the i_b and i_u are both sampled in the P_{global} since there is no P_{local}.

The second algorithm is the SAEA-LS which has the P_{global} and the P_{local}. The initial P_{global} and P_{local} are the same population generated by the Latin hypercube sampling. The local search adopts the proposed local search method in the Equation (11), and selection operator adopts the feasibility ranking. In addition, the global search uses the DE, and selection operator uses the stochastic ranking. For fair comparison, the other parameter settings are same as the SAEA.

Table 1: Optimal solutions obtained by the SAEA, SAEA-LS, and SAEA-RS on the 30D CEC2010 problems. The results are shown in the form of mean ± standard deviation.

Problem	SAEA	SAEA-LS	SAEA-RS
c01	$-1.96e-01 \pm 3.46e-02$	$-2.17e-01 \pm 3.44e-02$	$-2.18e-01 \pm 3.50e-02$
c07	$4.88e+09 \pm 4.86e+09$	$8.03e+09 \pm 5.99e+09$	$2.93e+08 \pm 2.62e+08$
c08	$7.34e+09 \pm 4.95e+09$	$1.09e+10 \pm 7.41e+09$	$3.16e+09 \pm 2.78e+09$
c13	$-2.28e+01 \pm 5.12e+00$	$-2.19e+01 \pm 4.14e+00$	$-3.79e+01 \pm 4.07e+00$
c14	56.67%	83.00%	70.00%
c15	3.33%	10.00%	3.33%

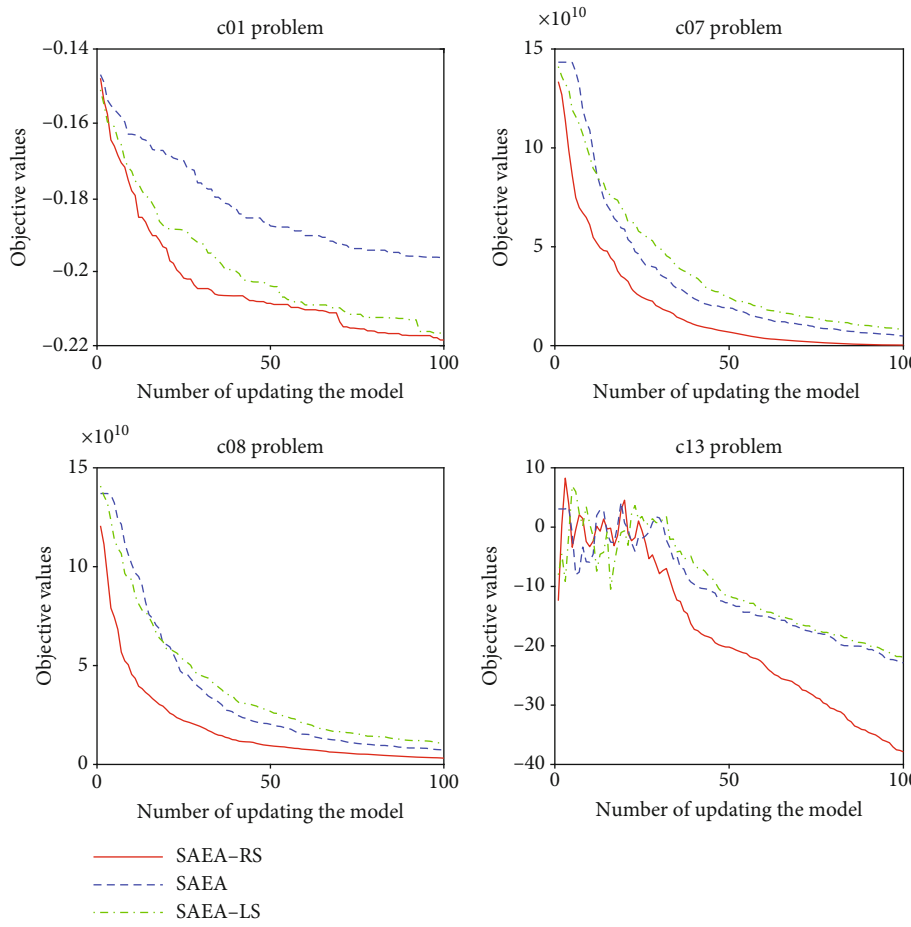

Figure 2: The convergence curves of SAEA, SAEA-LS, and SAEA-RS on c01, c07, c08, and c13 problems.

The third algorithm is the SAEA-RS. The only difference between SAEA-RS and SAEA-LS is that SAEA-RS adds the restart strategy proposed in Algorithm 1. The δ in the $P D_{\text{threshold}}$ is set to $1e$-10. The other parts including the parameter settings are same as SAEA-LS.

The ablation experiments adopts the 30D CEC2010 [29] as the benchmark problems. The equality constraint problems are not considered in this work, so the c01, c07, c08, c13, c14, and c15 are chosen as the test problems. To avoid the instability of the algorithms, the experiments are conducted independently for 30 times. The mean obtained optimum of 30 independent experiments are shown in Table 1. The best results among three algorithms are highlighted in bold. The average convergence curve on the c01, c07, c08, and c13 are shown in Figure 2.

From Table 1, SAEA-RS has achieved the best results on the c01, c07, c08, and c13 problems. The three algorithms have failed in finding the feasible solutions in some experiments on the c14 and c15 problems. Hence, the rate of finding the feasible solutions in 30 experiments are used as the comparison target. The SAEA-LS achieves the best results on c14 and c15 problems. In Figure 2, the SAEA-RS has the fastest convergence speed and best final results in c01, c07, c08, and c13 problems. Due to the restart strategy, the

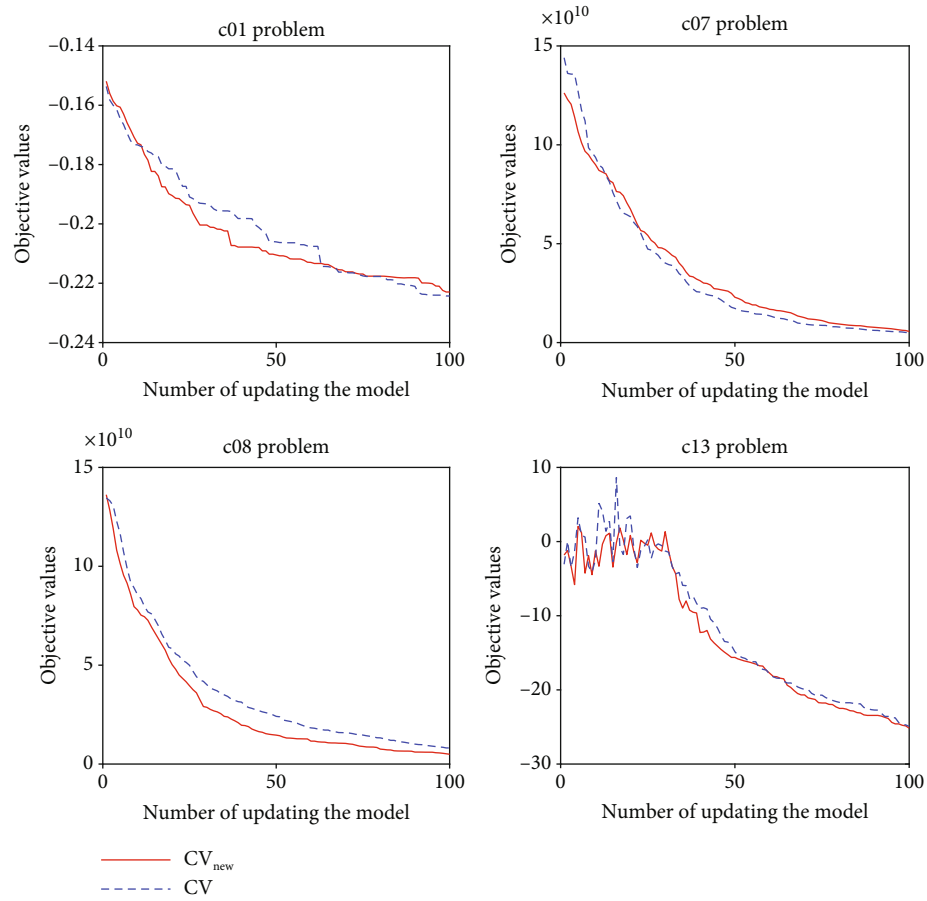

Figure 3: The convergence curves of SAEA-RS with CV and SAEA-RS with CV_{new} on c01, c07, c08, and c13 problems.

Table 2: Optimal solutions obtained by the SAEA-RS, GLoSADE, and MPMLS on the CEC2006 problems. The results are shown in the form of mean ± standard deviation.

Problem	SAEA-RS	GLoSADE	MPMLS
g01	−1.50e + 01 ± 1.09e − 09	−1.50e + 01 ± 2.42e − 02	−6.26e + 00 ± 8.76e − 01
g02	−2.08e − 01 ± 4.99e − 02	−2.40e − 01 ± 4.18e − 02	−2.46e − 01 ± 4.37e − 02
g04	−3.07e + 04 ± 7.96e − 08	−3.07e + 04 ± 1.19e + 00	−3.05e + 04 ± 5.25e + 01
g06	−6.96e + 03 ± 2.81e − 03	−6.86e + 03 ± 4.21e + 02	6.67%
g07	2.46e + 01 ± 1.69e − 01	2.81e + 01 ± 3.03e + 00	86.67%
g08	−8.21e − 02 ± 2.68e − 02	−9.58e − 02 ± 2.73e − 05	−7.86e − 02 ± 1.90e − 02
g09	1.19e + 03 ± 2.24e + 02	1.16e + 03 ± 2.65e + 02	9.43e + 02 ± 8.33e + 01
g10	7.32e + 03 ± 1.82e + 02	9.03e + 03 ± 8.09e + 02	93.33%
g12	−9.56e − 01 ± 3.20e − 02	−9.94e − 01 ± 1.20e − 02	−9.95e − 01 ± 6.07e − 03
g16	−1.90e + 00 ± 1.57e − 02	−1.63e + 00 ± 1.50e − 01	96.67%
g18	−7.83e − 01 ± 8.42e − 02	46.67%	0.00%
g19	1.18e + 02 ± 4.45e + 01	2.08e + 02 ± 9.63e + 01	8.37e + 02 ± 1.24e + 02
g24	−5.51e + 00 ± 4.65e − 11	−5.51e + 00 ± 4.97e − 03	−5.50e + 00 ± 2.96e − 03

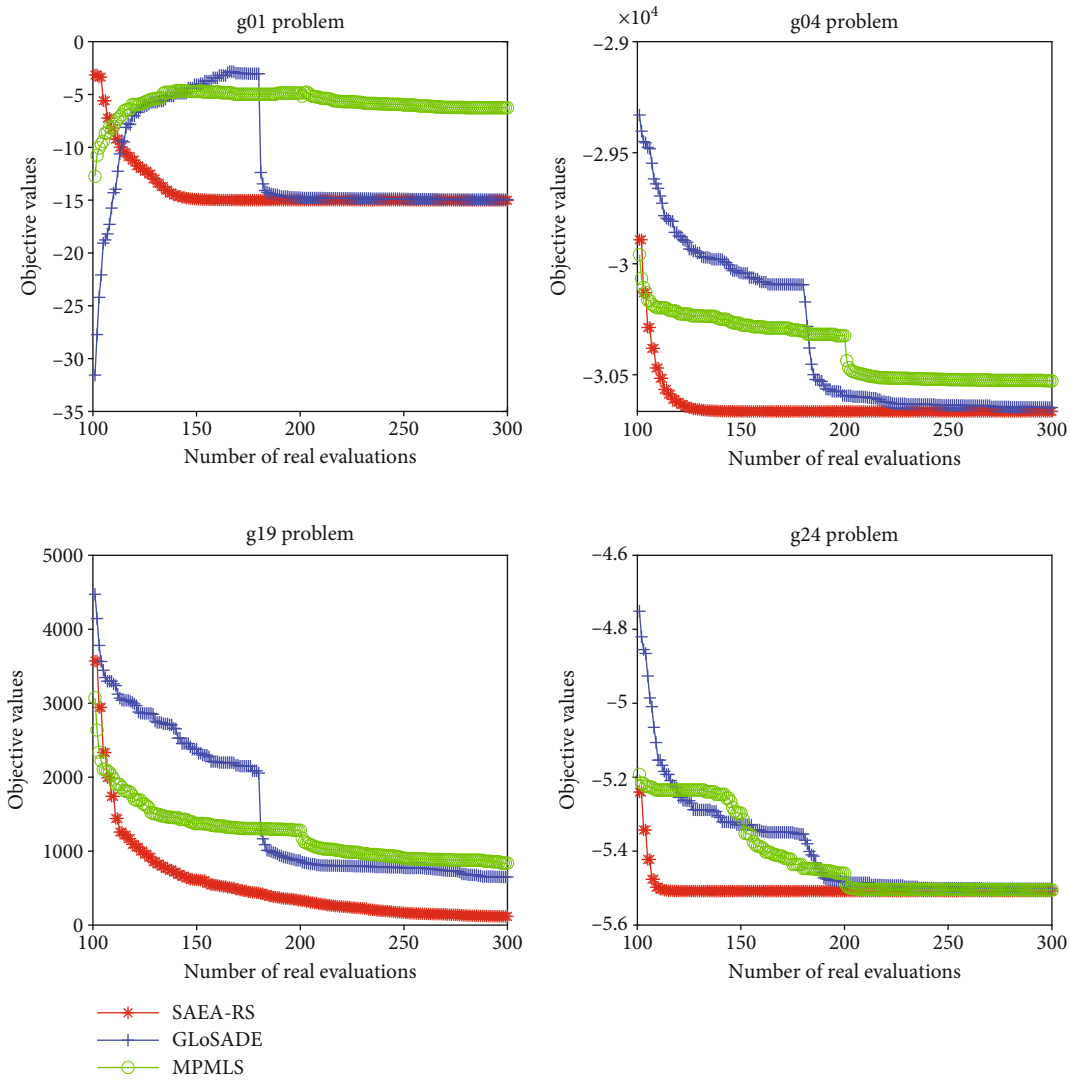

Figure 4: The convergence curves of SAEA-RS, GLoSADE, and MPMLS on g01, g04, g19, and g24 problems of CEC2006.

SAEA-RS has the faster convergence speed than the SAEA and SAEA-LS. Since the i_b is sampled in the P_{local}, the diversity of SAEA-LS is not as good as that of SAEA, and the performance of SAEA-LS is worse than the SAEA. The initial convergence curve has some fluctuations owing to not finding the infeasible solutions. In a word, the SAEA-RS performs best in the three algorithms.

For investigating the effects of the proposed CV_{new}, the SAEA-RS with CV_{new} is compared with the SAEA-RS with the general CV on the c01, c07, c08, and c13 problems of the 30D CEC2010. The other parameter settings are same as those in the previous experiments. The comparative experiments are carried out 30 times independently. The average convergence curves of the two algorithms are shown in Figure 3.

From Figure 3, the CV_{new} has a higher convergence speed than the CV on the c01, c08, and c13 problems. The CV_{new} has the lower convergence speed but similar final results on the c07 problems. Therefore, the CV_{new} has the different effects on the different problems. On the whole, the CV_{new} is benefit for improving the performance of the proposed algorithm, which can increase the convergence speed on some problems.

4.2. Comparative Experiment. In order to compare the performance of the proposed algorithm with other existing algorithms, the comparative experiment is carried out in this part. The SAEA-RS is compared with two state-of-the-art algorithms which are the GLoSADE and the MPMLS. For a fair comparison, the total real function evaluations of all algorithms are set to 300. The other parameter settings of the SAEA-RS are same as the ablation experiment. The rest of the parameter settings of GLoSADE and MPMLS are consistent with that in [22, 23]. The comparative experiments are, respectively, conducted on three sets of benchmark problems which are the CEC2006 [30], the CEC2010, and the CEC2017 [31]. To avoid the experimental instability, the comparative experiments are repeated 30 times

Table 3: Optimal solutions obtained by the SAEA-RS, GLoSADE, and MPMLS on the 30D CEC2010 problems. The results are shown in the form of mean ± standard deviation.

Problem	SAEA-RS	GLoSADE	MPMLS
c01	−2.25e − 01 ± 3.24e − 02	−1.90e − 01 ± 2.11e − 02	−1.86e − 01 ± 1.85e − 02
c07	4.01e + 08 ± 7.07e + 08	3.45e + 08 ± 4.53e + 08	2.14e + 10 ± 9.64e + 09
c08	3.50e + 09 ± 3.10e + 09	3.72e + 08 ± 3.41e + 08	1.80e + 10 ± 7.91e + 09
c13	−3.68e + 01 ± 5.88e + 00	−2.42e + 01 ± 5.59e + 00	0.00%
c14	90.00%	8.52e + 11 ± 1.35e + 12	93.33%
c15	13.33%	23.33%	10.00%

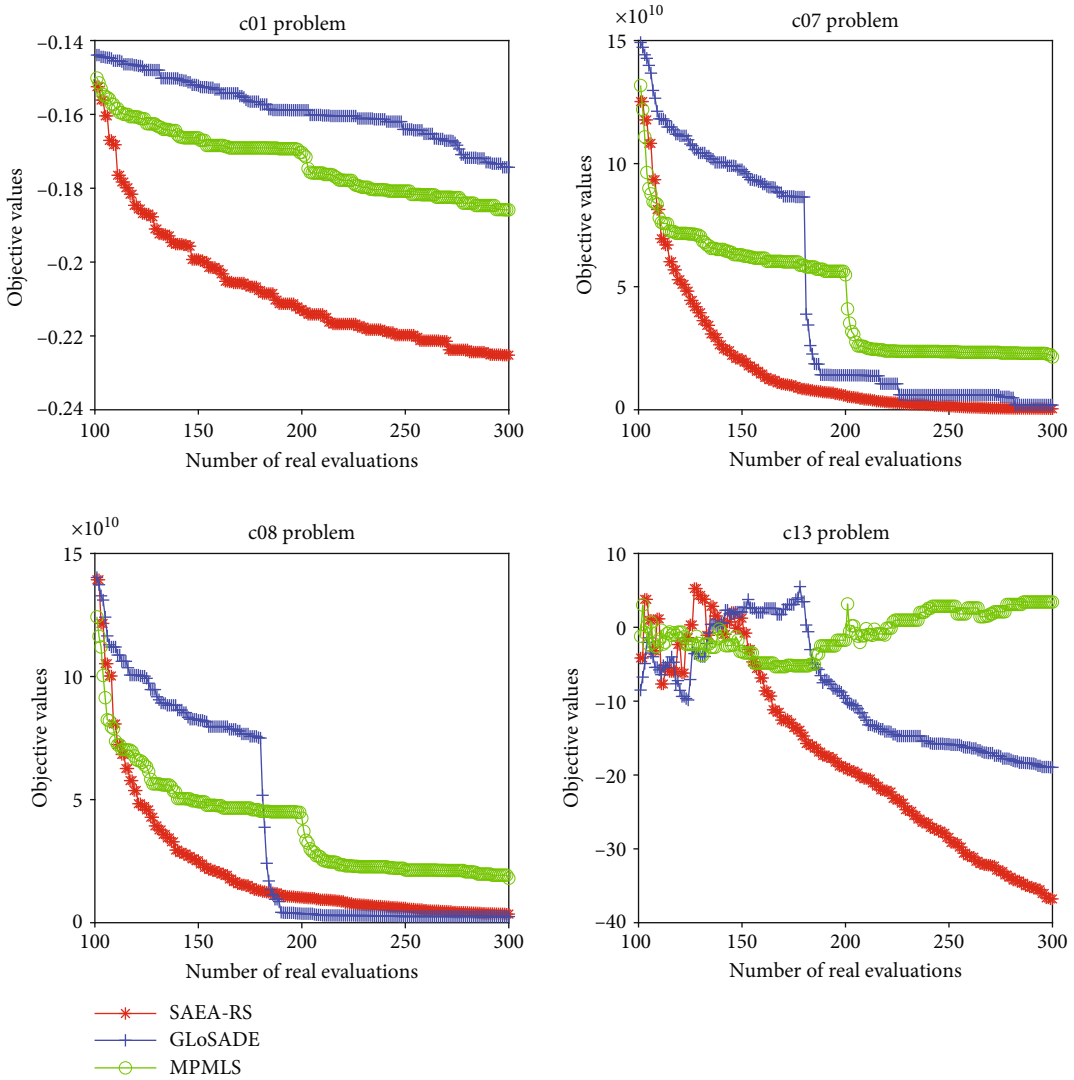

Figure 5: The convergence curves of SAEA-RS, GLoSADE, and MPMLS on c01, c07, c08, and c13 problems of CEC2010.

independently. The average optimal objective values of 30 experiments are as the performance indicator. If an algorithm fails to find a feasible solution in some experiments, the percentage of the algorithm finding a feasible solution in 30 experiments is used as the performance indicator.

The experimental results on 13 benchmark problems of CEC2006 are shown in Table 2, and the best results among all the algorithms are italicized. From Table 2, SAEA-RS and GLoSADE have acquired the very close results on the g01, g04, and g24 problems. However, SAEA-RS converges

Table 4: Optimal solutions obtained by the SAEA-RS, GLoSADE, and MPMLS on the 30D CEC2017 problems. The results are shown in the form of mean ± standard deviation.

Problem	SAEA-RS	GLoSADE	MPMLS
c01	6.77e + 04 ± 1.72e + 04	7.61e + 04 ± 2.04e + 04	9.11e + 04 ± 2.00e + 04
c02	2.50e + 04 ± 6.31e + 03	73.33%	3.79e + 04 ± 1.39e + 04
c04	4.17e + 02 ± 3.19e + 01	5.32e + 02 ± 9.25e + 01	5.49e + 02 ± 5.52e + 01
c05	8.85e + 03 ± 3.91e + 03	93.33%	80.00%
c13	0.00%	0.00%	0.00%
c19	0.00%	0.00%	0.00%
c20	1.14e + 01 ± 7.30e − 01	1.10e + 01 ± 8.91e − 01	1.12e + 01 ± 6.39e − 01
c22	0.00%	0.00%	0.00%
c28	0.00%	0.00%	0.00%

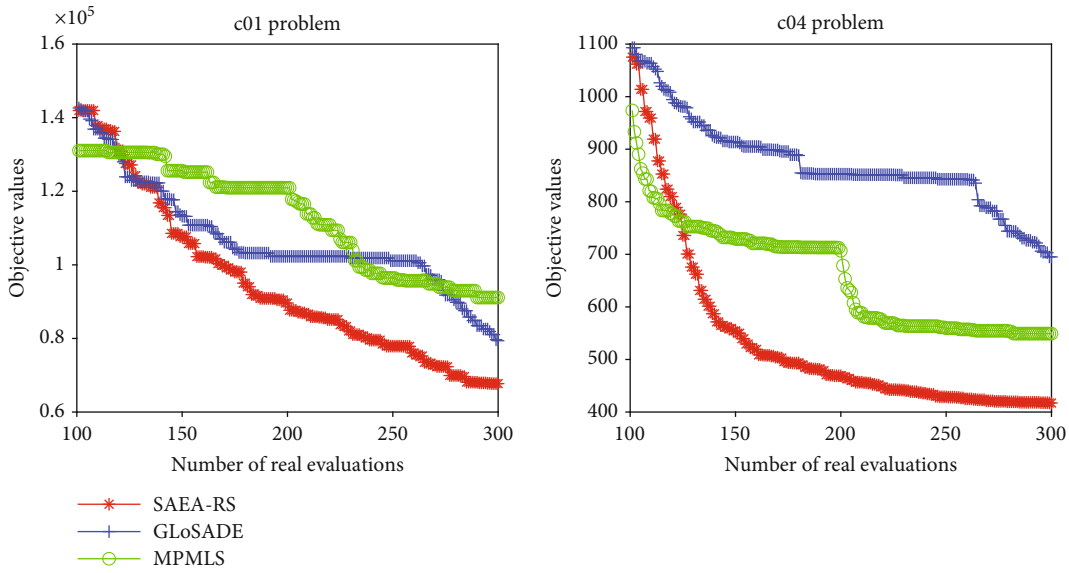

Figure 6: The convergence curves of SAEA-RS, GLoSADE, and MPMLS on c01 and c04 problems of CEC2017.

faster and achieves more accurate results than GLoSADE. On g06, g07, g10, g16, g18, and g19 problems, SAEA-RS achieves a significant lead. Especially, SAEA-RS obtains feasible solutions in all 30 experiments on g18 problems, but GLoSADE and MPMLS do not obtain the feasible solutions in some experiments. On the whole, SAEA-RS gets a lead on 9 of the 13 benchmark problems. In order to explore the convergence performance of the algorithms, the convergence curve of SAEA-RS, GLoSADE, and MPMLS on g01, g04, g19, and g24 problems are shown in Figure 4. The vertical axis is the objective values of the current best solution, and the horizontal axis is the number of real evaluations. Since the first 100 real evaluations are applied in the initial sampling, the horizontal axis is set from 100 to 300. The objective values of GLoSADE and MPMLS increase in the first 200 real evaluations owing to not finding the feasible solutions, while SAEA-RS converges rapidly and even reaches near the optimum. The GLoSADE converges to the optimum rapidly at the 200 real evaluations since GLoSADE enters the local search stage, which have the similar situations in other problems. Similarly, SAEA-RS has the fastest convergence speed among the three algorithms on g04, g19, and g24 problems. Therefore, when the number of real evaluations is small, the advantages of SAEA-RS are more prominent and more suitable for expensive problems than GLoSADE and MPMLS.

The experiment results on 30D CEC2010 are shown in Table 3. SAEA-RS acquires the best results on c01 and c13 problems, and GLoSADE acquires best results of the remaining four problems. However, the convergence curves in Figure 5 show that SAEA-RS has the best performance on the convergence speed. SAEA-RS has the best results on c01, c07, c08, and c13 problems before the 200 real evaluations. Owing to the local search, GLoSADE converges quickly after 200 real evaluations on c07 and c08 problems, but the final results are very close to SAEA-RS. In particular, the convergence curve fluctuates on c13 problems since SAEA-RS and GLoSADE do not find the feasible solutions before 200 real evaluations and MPMLS do not find the feasible solutions in 300 real evaluations.

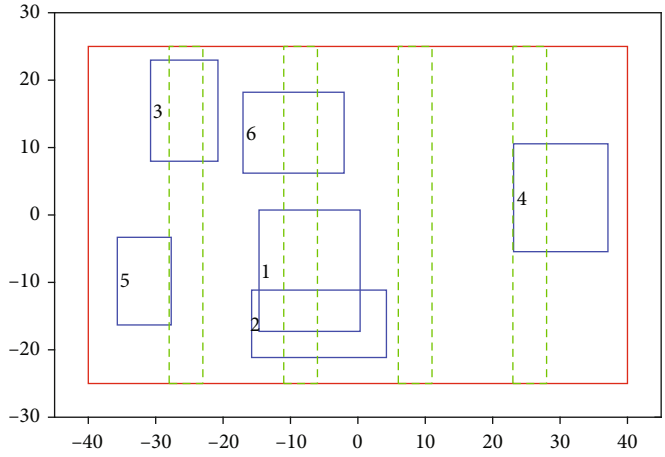

Figure 7: The schematic diagram of component layout.

Input: P_{lhsample}: the population generated by Latin hypercube sampling, max_iter: the iterations
Output: P_{local}: the local population, P_{global}: the global population, $M_{\mathscr{H}_{\max}}$: the $\mathscr{H}_{\max}$ objective model, $M_{\mathscr{G}_{\text{heat}}}$: the $\mathscr{G}_{\text{heat}}$ constraint model, D_{train}: the training dataset.
1 Calculate $\mathscr{G}_{\text{overlap}}(\mathbf{x})$, $\mathscr{G}_{\text{centroid}}(\mathbf{x})$ and $\mathscr{G}_{\text{pipe}}(\mathbf{x})$ of each individual $\mathbf{x}$ in the P_{lhsample} and take the sum of them as the fitness f.
2 Calculate $\mathscr{H}_{\max}(\mathbf{x})$ and $\mathscr{G}_{\text{heat}}(\mathbf{x})$ of each individual $\mathbf{x}$ in the P_{lhsample} by the thermodynamic simulations and take them as the training dataset D_{train}.
3 Train $M_{\mathscr{H}_{\max}}$ and $M_{\mathscr{G}_{\text{heat}}}$ with the D_{train}.
4 Set $A = \varnothing$.
5 **for** $i = 1 : \max\_\text{iter}$ **do**
6 **for** $j = 1 : |P_{\text{lhsample}}|$ **do**
7 Generate a new individual $\mathbf{x}_{\text{new}}$ by Equation (11) where $\mathbf{x}_{r1}$ is the j_{th} individual $\mathbf{x}_j$ in P_{lhsample}, $\mathbf{x}_{r2}$ and $\mathbf{x}_{r3}$ are two individuals randomly selected from P_{lhsample}.
8 Calculate the fitness f_{new}(*The sum of* $\mathscr{G}_{overlap}(\boldsymbol{x}_{new})$, $\mathscr{G}_{centroid}(\boldsymbol{x}_{new})$ *and* $\mathscr{G}_{pipe}(\boldsymbol{x}_{new})$) of the $\mathbf{x}_{\text{new}}$.
9 **if** $f_{\text{new}} == 0$ **then**
10 $A = A \cup \mathbf{x}_{\text{new}}$
11 end
12 **if** $f_{\text{new}} \le f_j$ **then**
13 $\mathbf{x}_j = \mathbf{x}_{\text{new}}$
14 $f_j = f_{\text{new}}$
15 end
16 end
17 end
18 **if** $|A| \ge |P_{\text{lhsample}}|$ **then**
19 A is clustered into $|P_{\text{lhsample}}|$ classes.
20 Select the individuals nearest to the each cluster center and take them as P_{local}.
21 else
22 Keep searching for new individuals until $|A| \ge |P_{\text{lhsample}}|$.
23 end
24 Set P_{global} as P_{local}.

Algorithm 3: Pseudocode of population initialization.

The average results of SAEA-RS, GLoSADE, and MPMLS on 9 benchmark problems of 30D CEC2017 are reported in Table 4. SAEA-RS achieves the best results on c01, c02, c04, and c05 problems, and GLoSADE acquires the best solutions on c20 problems. In addition, all the algorithms fail to find a feasible solution in 30 experiments on the c13, c19, c22, and c28 problems. Therefore, SAEA-RS has an outstanding performance on the 30D CEC2017 problems. The convergence curve on c01 and c04 problems are shown in Figure 6. SAEA-RS has the fastest

Table 5: Optimal solutions obtained by the SAEA-RS on the CLOP of 10 independent repeated experiments.

No.	x1	y1	x2	y2	x3	y3	x4	y4	x5	y5	x6	y6	$\mathscr{H}_{\max}$	CV_{new}
1	−3.22e + 01	1.17e + 01	1.60e + 01	6.53e − 01	3.30e + 01	1.75e + 01	2.72e − 01	−1.39e + 01	8.67e + 00	1.82e + 01	3.20e + 01	−1.61e + 01	2.55e + 01	5.07e − 01
2	−1.57e + 01	5.12e + 00	3.00e + 01	−6.94e + 00	2.38e + 00	−6.66e + 00	−1.78e + 01	−1.64e + 01	1.24e + 01	8.76e + 00	3.04e + 01	1.50e + 01	2.30e + 01	0.00e + 00
3	−9.59e + 00	1.08e + 01	4.36e + 00	−3.50e + 00	2.90e + 01	8.78e + 00	−2.50e + 01	−1.29e + 01	2.43e + 01	−1.24e + 01	8.31e + 00	1.42e + 01	2.60e + 01	1.00e + 00
4	−2.66e + 01	1.39e + 01	2.29e + 01	−1.90e + 01	−9.75e − 01	−1.56e + 01	−8.31e − 01	7.86e + 00	−1.23e + 01	−1.62e + 01	1.62e + 01	3.28e + 00	2.00e + 01	5.25e − 01
5	2.09e + 01	−1.10e + 01	3.93e + 00	−1.73e + 01	1.03e + 01	1.67e + 01	−3.11e + 01	−1.09e + 01	−3.24e + 01	1.10e + 01	−1.89e + 00	1.41e + 01	2.60e + 01	1.00e + 00
6	−2.67e + 01	5.33e + 00	9.35e + 00	2.00e + 01	3.04e + 01	1.69e + 01	−7.53e + 00	−1.52e + 01	−1.87e + 01	−1.79e + 01	1.82e + 01	1.19e + 00	2.60e + 01	1.00e + 00
7	2.14e + 01	−5.23e + 00	−4.58e + 00	2.23e + 00	1.03e + 01	1.68e + 01	−2.80e + 01	−1.33e + 01	4.59e + 00	−1.41e + 01	−1.04e + 01	1.35e + 01	2.60e + 01	1.00e + 00
8	−5.82e − 01	1.30e + 01	2.97e + 01	−1.83e + 01	−3.25e + 01	1.09e + 01	−1.41e − 01	−1.49e + 01	−3.01e + 01	−1.03e + 01	3.21e + 01	1.26e + 01	2.30e + 01	0.00e + 00
9	8.73e + 00	6.42e + 00	−2.96e + 01	−1.33e + 01	2.21e + 01	−7.27e + 00	−1.38e + 01	1.62e + 01	3.06e + 01	−1.35e + 01	−2.72e + 01	4.26e − 01	2.60e + 01	1.00e + 00
10	−2.66e + 01	5.06e + 00	2.40e + 01	1.96e + 01	1.25e + 01	−6.03e + 00	−1.26e + 01	−4.40e + 00	3.44e + 00	8.78e + 00	2.86e + 01	−1.20e + 01	2.60e + 01	1.00e + 00

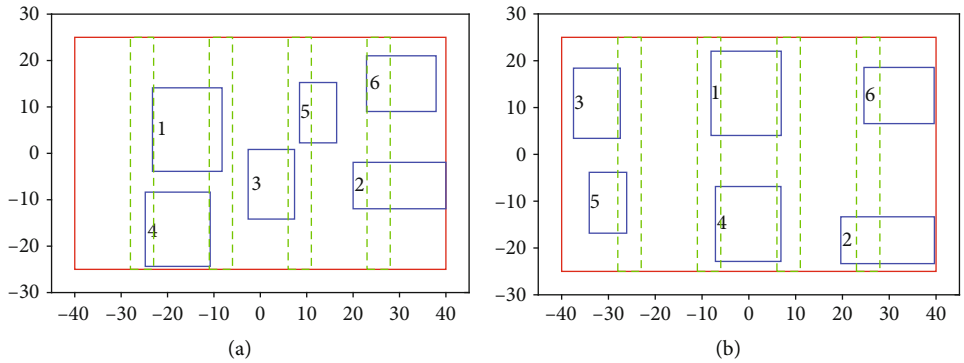

Figure 8: The optimal component layouts of solution no. 2 and no. 8.

convergence speed on c01 and c04 problems which reflects the advantage of SAEA-RS when the real evaluations is small.

In summary, SAEA-RS has the highest convergence speed and finds the best solutions on most benchmark problems. For the constrained expensive optimization problems, the advantages of SAEA-RS can help to find a better solution within a few real evaluations.

5. Application to Electronic Component Layout Optimization

In this section, the proposed algorithm is applied to solve a CLOP on a space circuit board. As shown in Figure 7, the red border indicates the circuit board whose length is 80, and the width is 50. The four green rectangles represent the heat pipes on the circuit board, and the blue rectangles represent 6 different electronic components. The heat pipes can evacuate the heat generated by the components. The length and width of component 1 are 15 and 18, respectively, which can be recorded as (15, 18). Similarly, the size of component 2 to 6 can be denoted as (20, 10), (10, 15), (14, 16), (8, 13), and (15, 12). The first constraint is that the components should not overlap each other or exceed the layout domain. The second constraint is that the deviation between the centroid of the system and the expected centroid should not exceed a specific value. The third constraint is that the heat load on each heat pipe should not exceed its maximum capacity. The last constraint is that each component should overlap with the heat pipes to dissipate heat. For uniform heat distribution, the objective is to minimize the maximum thermal load on all heat pipes. The problem can be described as follows:

$$\begin{aligned} &\text{minimize} && \mathscr{H}_{\max}(\mathbf{x}), \\ &\text{subjectto} && \mathscr{G}_{\text{overlap}}(\mathbf{x}) \le 0, \quad \mathscr{G}_{\text{centroid}}(\mathbf{x}) \le 0, \\ &&& \mathscr{G}_{\text{heat}}(\mathbf{x}) \le 0, \quad \mathscr{G}_{\text{pipe}}(\mathbf{x}) \le 0, \end{aligned} \tag{12}$$

where $\mathscr{H}_{\max}$ denotes the maximum thermal load, $\mathscr{G}_{\text{overlap}}$ denotes the component layout overlap constraint, $\mathscr{G}_{\text{centroid}}$ denotes the centroid deviation constraint, $\mathscr{G}_{\text{heat}}$ denotes the heat load constraint, $\mathscr{G}_{\text{pipe}}$ denotes the heat pipe overlap constraint, and $\mathbf{x} = [x_1, y_1, \cdots, x_6, y_6]$, x_i and $y_i (i = 1, \cdots, 6)$ denotes the center coordinate of component 1 to 6, $-40 \le x_i \le 40$, and $-25 \le y_i \le 25$.

The $\mathscr{H}_{\max}$ and $\mathscr{G}_{\text{heat}}$ require the thermodynamic simulation which is expensive and time-consuming, but $\mathscr{G}_{\text{overlap}}$, $\mathscr{G}_{\text{centroid}}$, and $\mathscr{G}_{\text{pipe}}$ can be easily calculated by the cheap function evaluations. In order to take advantage of this characteristic, a population initialization strategy shown in Algorithm 3 is designed for replacing the original Latin hypercube sampling in the SAEA-RS. Through Algorithm 3, there are more approximately feasible individuals selected as the initial P_{local} and P_{global}, which can increase the convergence process to some extent.

In this section, the max_iter is set to 1000. The other parameter settings of SAEA-RS are same as those in Section 4. Therefore, there are 300 real thermodynamic simulations used in the whole optimization process, which are 100 simulations for evaluating the $\mathscr{H}_{\max}$ and $\mathscr{G}_{\text{heat}}$ of the P_{lhsample} in Algorithm 3 and 200 simulations for evaluating the new sampling i_b and i_u in Algorithm 2. It should be noted that $\mathscr{G}_{\text{overlap}}$, $\mathscr{G}_{\text{centroid}}$, and $\mathscr{G}_{\text{pipe}}$ are always calculated by the cheap functions in the whole optimization process. The experiments are repeated independently for 10 times, and the final optimal results are shown in Table 5.

From Table 5, the $x1$ to $x6$ and $y1$ to $y6$ are the horizontal and vertical coordinates of six components' geometric centers, respectively. $\mathscr{H}_{\max}$ is the objective value, and CV_{new} is the constraint violation. The experiment no. 2 and no. 8 have found the feasible solutions. However, other experiments do not find the feasible solutions which shows that the feasible region of the CLOP is very small. To show more intuitive results, the component layout of solutions no. 2 and no. 8 are shown in Figure 8. The $\mathscr{H}_{\max}$ and the CV_{new} are both 23 and 0 of experiment no. 2 and no. 8. From Figure 8, components are uniformly distributed on the board and do not overlap each other. The overall centroid of the components is centered which guarantees the stability of the overall board. Also, the components are all in contact with heat pipe so that the components can dissipate the heat through the heat pipes. On the whole, the algorithm has realized the optimization requirements. The

proposed algorithm converges under the small number of thermodynamic simulations, which greatly speeds up the optimization process.

6. Conclusions

For addressing the expensive constrained optimization problems in space engineering, we propose a surrogate-assisted evolutionary algorithm with restart strategy. The local search has the strong exploitation ability, while the global search is on the exploration ability. Therefore, the global search and local search are combined to give full play to their advantages. A restart strategy is proposed to make the local search jump out of the local optimum promptly. In addition, a new local search method and a constraint violation calculation approach are proposed for improving the algorithm performance. The comparative experiments compare SAEA-RS with two state-of-the-art algorithms. The results demonstrate that SAEA-RS has a higher convergence speed and the better results on CEC2006, CEC2010, and CEC2017 problems under very limited computation budgets. Meanwhile, SAEA-RS is applied to solve the electronic component thermal layout optimization problems in space engineering. The final results show that the problem is well solved which indicates the great significance of SAEA-RS to solve real-world engineering problems.

There are several aspects worthy of further study in the future. Firstly, the restart strategy can be integrated in other algorithm frameworks for improving the population diversity. Secondly, other constraint handling methods can be applied in the proposed algorithm for further exploration. Thirdly, the proposed algorithm applied to other real-world problems needs further study.

Data Availability

The data used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Authors' Contributions

Lei Han worked on the algorithm and experiments, Handing Wang supervised this work, and Shuo Wang improves the writing.

References

[1] D. Dasgupta and Z. Michalewicz, *Evolutionary Algorithms in Engineering Applications*, Springer Science & Business Media, 2013.

[2] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, no. 11, pp. 1223–1241, 2002.

[3] Z. Li, Q. Zhang, X. Lin, and H.-L. Zhen, "Fast covariance matrix adaptation for large-scale black-box optimization," *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 2073–2083, 2018.

[4] D. E. Goldberg, "Genetic algorithms in search, Optimization, and Machine Learning," *Addion Wesley*, vol. 102, no. 1989, p. 36, 1989.

[5] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[6] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: an overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2019.

[7] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.

[8] Y. Jin, "Surrogate-assisted evolutionary computation: recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[9] T. Chugh, N. Chakraborti, K. Sindhya, and Y. Jin, "A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem," *Materials and Manufacturing Processes*, vol. 32, no. 10, pp. 1172–1178, 2017.

[10] 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, 2014.

[11] Z. Zhou, Y. S. Ong, M. H. Nguyen, and D. Lim, "A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm," in *2005 IEEE Congress on Evolutionary Computation*, pp. 2832–2839, Edinburgh, UK, 2005.

[12] I. Loshchilov, M. Schoenauer, and M. Sebag, "Comparison-based optimizers need comparison-based surrogates," in *International Conference on Parallel Problem Solving from Nature*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., vol. 6238 of Lecture Notes in Computer Science, pp. 364–373, Springer, Berlin, Heidelberg, 2010.

[13] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[14] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven con- strained multiobjective combinatorial optimization of trauma systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 536–549, 2020.

[15] H. Wang, Y. Jin, C. Sun, and J. Doherty, "Offline data-driven evolutionary optimization using selective surrogate ensembles," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 203–216, 2019.

[16] D. W. Coit, A. E. Smith, and D. M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," *Informs Journal on Computing*, vol. 8, no. 2, pp. 173–182, 1996.

[17] C. Yu, K. L. Teo, L. Zhang, and Y. Bai, "A new exact penalty function method for continuous inequality constrained optimization problems," *Journal of Industrial and Management Optimization*, vol. 6, no. 4, pp. 895–910, 2010.

[18] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311–338, 2000.

[19] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[20] C. A. C. Coello, "Treating constraints as objectives for single-objective evolutionary optimization," *Engineering Optimization+ A35*, vol. 32, no. 3, pp. 275–308, 2000.

[21] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 560–575, 2007.

[22] Y. Wang, D.-Q. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1642–1656, 2019.

[23] G. Li and Q. Zhang, "Multiple penalties and multiple local surrogates for expensive constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 769–778, 2021.

[24] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT Press, 1992.

[25] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, no. 3, pp. 193–212, 1995.

[26] T. Takahama, S. Sakai, and N. Iwane, "Constrained optimization by the ε constrained hybrid algorithm of particle swarm optimization and genetic algorithm," in *Australasian Joint Conference on Artificial Intelligence*, S. Zhang and R. Jarvis, Eds., vol. 3809 of Lecture Notes in Computer Science, pp. 389–400, Springer, Berlin, Heidelberg, 2005.

[27] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *Evolutionary Computation*, vol. 9, pp. 1–17, 2005.

[28] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[29] R. Mallipeddi and P. N. Suganthan, *Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization*, vol. 24, Nanyang Technological University, Singapore, 2010.

[30] J. Liang, T. P. Runarsson, E. Mezura-Montes et al., "Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, pp. 8–31, 2006.

[31] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization," Technical Report, National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, 2017.