

A Surrogate Model Assisted Estimation of Distribution Algorithm with Multi-acquisition Functions for Expensive Optimization

Hao Hao, Shuai Wang, Bingdong Li, and Aimin Zhou

Shanghai Frontiers Science Center of Molecule Intelligent Syntheses

Shanghai Institute of AI for Education, School of Computer Science and Technology

East China Normal University, Shanghai, China

52194506007@stu.ecnu.edu.cn wangshuai515658@163.com {bdli, amzhou}@cs.ecnu.edu.cn

Abstract—The estimation of distribution algorithm (EDA) is an efficient heuristic method for handling black-box optimization problems since the ability for global population distribution modeling and gradient-free searching. However, the trial and error search mechanism relies on a large number of function evaluations, which is a considerable challenge under expensive black-box problems. Therefore, this article presents a surrogate assisted EDA with multi-acquisition functions. Firstly, a variable-width histogram is used as the global distribution model that focuses on promising areas. Next, the evaluated-free local search method improves the quality of new generation solutions. Finally, model management with multiple acquisitions maintains global and local exploration preferences. Several commonly used benchmark functions with 20 and 50 dimensions are adopted to evaluate the proposed algorithm compared with several state-of-the-art surrogate assisted evaluation algorithms (SAEAs) and Bayesian optimization method. In addition, a rover trajectories optimizing problem is used to verify the ability to solve complex problems. The experimental results demonstrate the superiority of the proposed algorithm over these comparison algorithms.

Index Terms—Surrogate assisted Evolutionary Algorithm, Estimation of distribution algorithm, Acquisition functions

I. INTRODUCTION

Global optimization of black-box and non-convex functions is an important component of real-world application [1] and machine learning task [2]. From structural optimization of industrial manufacturing to policy search for reinforcement learning, these optimization problems highlight two important features: The first is the absence of closed form expressions. When internal workings are difficult to trace, the sophisticated model might be considered as a black-box because it can only be viewed in terms of its inputs and outputs [3]. The second is the high cost of optimization. The evaluation of black-box functions is accompanied by computer simulation experiments, physical simulations, etc. This approach necessitates domain expertise as well as a series of experiments, which may be both costly and time demanding.

Automated optimization algorithms such as Bayesian optimizations (BOs) [4] and surrogate assisted evolution algorithms (SAEAs) [5] have been introduced, which construct the

relationship between parameter combinations and the response of black-box functions. The BO usually builds probabilistic models using the Gaussian processes for black-box functions and uses the acquisition function to refine the posterior distribution of the model, striking a balance between model accuracy and optimal solution search. Surrogate-assisted, also known as metamodel-assisted evolutionary, has received increasing attention in recent years. Heuristic reproduction operators are employed in SAEAs to generate promising solutions. SAEAs build surrogate models based on the evaluated solutions and predict new trial solutions instead of the original objective functions. Most of these algorithms can be classified into three categories according to the types of surrogates.

Regression based SAEAs are the most widely used form. Regression models capture mapping from a decision variable vector \mathbf{x} to its corresponding objective value $f(\mathbf{x})$. The most commonly used surrogate models include polynomial regression (PR) [6] also referred as response surface methodology [7], radial basis function (RBFs) networks [8] and Gaussian processes (GPs) [9], also known as to Kriging [10]. Specifically, Liu et al. [9] proposed a GPs assisted differential evolution (DE) to solve the expensive problem in which dimension reduction methods are utilized to reduce the dimension of feature size of the training dataset. The RBFs were proposed by Sun et al. [8] to assist cooperative swarm optimization of high-dimensional expensive problems.

Classification based SAEAs are another common paradigm in which models are learned the labels of solutions according to their quality. Support vector machines (SVMs) [11], artificial neural network (ANNs) [12] and fuzzy K-nearest neighbor (FKNN) [13] are used as for surrogate. Classifiers are used by a memetic algorithm (MA) [14] to estimate the feasibility boundary. SVMs and FKNN are used to assist the DE in identifying the solution with ‘promising’ categories and rejecting ‘unpromising’ solutions.

The third is a new class of modeling methods called relation based SAEAs. As stated in [15], EAs are comparison-based algorithms. Therefore, they can only use comparisons between objective values and not the values themselves to drive the evolving process. Thus the relational model is used to learn

This work is supported by the Fundamental Research Funds for the Central Universities.

comparative relationships between solutions. The previously mentioned supervised learning methods can all be used to learn relations. However, the core difference is that the model is used to predict relation (superiority or inferiority) between two solutions, either the fitness value or the category of a solution. In [16], [17], surrogate models are used to predict the ranks of solutions that make full use of the flexibility inherent in the ordering of population. In [18], the convolutional neural network (CNN) is proposed to approximate the difference of fitness function values of two solutions rather than directly approximate the fitness of a solution. Recently, Hao et al. [19] proposed a preselection framework to learn the relation between pairs of solutions.

The previous studies on SAEAs mainly focus on the surrogate model designed. Moreover, the generation of new solutions relies on the differential evolution (DE), genetic algorithm (GA), and particle swarm optimization (PSO) while ignoring another type of efficient heuristic algorithm EDAs.

This paper proposed a surrogate model assisted EDA (SA-EDA). The variable-width histogram (VWH) is used for new solution generation, providing efficient search capabilities on problems with different scales and landscapes. Three acquisition functions (model management strategies) are used to guide the sampling, overcoming the instability of a single acquisition function. The contributions are as follows.

- The multi acquisition functions guiding SAEA with variable-width histogram model are introduced, named SA-EDA, for the black-box optimization problem.
- Two surrogate models are applied in SA-EDA for small- and medium-scale problems. The experimental studies demonstrate its advantages.

The rest of the paper is outlined as follows. In Section II, an introduction to related work is given. Next, the details of the main framework are described in Section III. Section IV empirically studies SA-EDA in small- and medium-scale problems on two widely used test suites. Finally, Section V concludes the paper with some future research remarks.

II. PRELIMINARIES

A. Black-box Optimization

An unconstrained minimization black-box problem can be formulated as follow:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad (1)$$

Where $\mathbf{x} = (x_1, \dots, x_n)^T$ is a decision variable vector, $\Omega \in R^n$ defines the feasible region of the search space, and $f : R^n \rightarrow R$ is the objective function, which is black-box since it can only be viewed in terms of its inputs and outputs when internal workings are hard to track. In some real-world applications, $f(\cdot)$ is costly expensive. Due to their expensive cost with no close form objective function poses a great challenge to traditional numerical and heuristic optimization.

B. Surrogate model

This paper uses Gaussian process (GP) [20] as the surrogate model for black-box functions. For a black-box function (1), GP modeling assumes that $f(\mathbf{x})$ at any point \mathbf{x} is a Gaussian random variable $\mathcal{N}(\mu, \sigma^2)$, where μ and σ are two constants independent of \mathbf{x} . For any \mathbf{x} , $f(\mathbf{x})$ is sample from a unknown function that are fully specified by a mean function and a covariance function with smoothness assumptions on $f(\cdot)$. For the dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, one unevaluation point \mathbf{x} , The best linear unbiased predictor of $f(\mathbf{x})$ is

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T C^{-1} (f^N - \mathbf{1}\hat{\mu}) \quad (2)$$

the uncertainty is

$$\sigma(\mathbf{x}) = \sqrt{\hat{\sigma}^2 \left[1 - \mathbf{r}^T C^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}^T C^{-1} \mathbf{r})^2}{\mathbf{1}^T C^{-1} \mathbf{1}} \right]} \quad (3)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are hyper parameter estimated by maximizing the likelihood of dataset \mathcal{D} . Let $c(x_i, x_j)$ is the correlation between x_i and x_j . Then C is a $N \times N$ matrix whose (i, j) -element is $c(x_i, x_j)$. $\mathbf{r} = (c(x, x_1), \dots, c(x, x_N))^T$. $\mathbf{1}$ is a N -dimensional column vector of ones.

C. Estimation of Distribution Algorithm

Estimation of distribution algorithms (EDAs) [21] is a special paradigm of EAs that uses probabilistic models to guide the population search rather than traditional reproduction operators such as crossover or mutation. Specifically, in EDAs, a Bayesian network often represents variable linkages obtained through various machine learning and statistical learning techniques. For EDAs, the prerequisite for high-quality sampling is a suitable Bayesian network. Based on variable linkages, Bayesian network models in EDAs can be summarized in three classes [21].

- **Univariate Models:** In this category, decision variables are assumed to be independent. The joint probability of the variables is fully factorized as

$$P(\mathbf{x}) = \prod_{j=1}^n P(x_j) \quad (4)$$

where x_j is j th variable of \mathbf{x} .

- **Bivariate Models:** Pairwise variable interactions are considered as

$$P(\mathbf{x}) = \prod_{j=1}^n P(x_j | x_{\pi_j}) \quad (5)$$

where x_{π_j} is the single parent variable of x_j .

- **Multivariate Models:** Multivariate interactions are considered as

$$P(\mathbf{x}) = \prod_{j=1}^n P(x_j | \Pi_j) \quad (6)$$

where Π_j is a set of parent variables of x_j .

In traditional EAs, the new trial solutions are generated by crossover and mutation operators. The major components are directly used by their parents. Therefore, the location information of parent solutions is directly used. The new solutions are close to their parents but far away from promising solutions. In contrast, EDA works with probabilistic models. At the same time, the global distribution information of the population is used to guide new solutions generation. So, local characteristics of populations are not well captured. Considering the nature of EDA, some combined methods are designed to improve the efficiency of EAs [21].

III. PROPOSED ALGORITHM

This section shows a surrogate model assisted EDA with a multi-acquisition function guided search. We first introduced the main framework of SA-EDA. Next, the implementation details of EDA with the local search are presented. Finally, a multi-acquisition function guided selection strategy is introduced.

A. Framework Work

We would like to make the following comments on Algorithm 1

Algorithm 1: Pseudocode of Main Framework

Input : N_0 (number of sampling in initiation);
 $maxFEs$ (maximum number of FEs);
 N (population size in evolution);
 α (number of surrogate model training set);
 t_{max} (number of local iterate on surrogate);

```

1 Initialize archive  $\mathcal{A} \leftarrow LHS(N_0)$ .
2  $fe \leftarrow N_0$ .
3 while  $fe \leq maxFEs$  do
4    $\mathcal{A} \leftarrow \text{Sort}(\mathcal{A}, \text{'ascend'})$ .
5    $\mathcal{D} \leftarrow \mathcal{A}_{1:\alpha}$ .
6    $\mathcal{P} \leftarrow \mathcal{A}_{1:N}$ .
7   Fit a surrogate model  $\mathcal{M}$  to the dataset  $\mathcal{D}$ .
8   for  $t \leftarrow 1$  to  $t_{max}$  do
9      $\mathcal{Q} \leftarrow \text{Generate\_solutions}(\mathcal{P})$ 
10     $\mathcal{P} \leftarrow \text{SA\_selection}(\mathcal{Q} \cup \mathcal{P}, \mathcal{M})$ 
11  end
12   $\mathcal{Q}_{best} \leftarrow \text{SA\_selection}(\mathcal{Q}, \mathcal{M}, \text{'hedge'})$ 
13   $\mathcal{A} \leftarrow \mathcal{A} \cup \text{Evaluation}(\mathcal{Q}_{best})$ .
14   $fe \leftarrow fe + 1$ .
15 end

```

Output: \mathcal{A}_{best} (the optimum of the solution).

- 1) **Initialization (lines 1-2):** N_0 initial solutions are sampled from the search space Ω by using the Latin hypercube sampling method [22]. Each solution is evaluated by black-box function $f(\cdot)$ and to from the archive \mathcal{A} . Then, update the evaluated count fe .
- 2) **Stopping condition (line 3):** The algorithm stops when the number of black-box function evaluation exceeds the given maximum FE_{max} .

- 3) **Prepare data (lines 4-6):** Arrange \mathcal{A} in ascending order according to their objective value and set the population \mathcal{P} and surrogate model training data \mathcal{D} .
- 4) **Model building (line 7):** The surrogate model \mathcal{M} is trained based on the the data set \mathcal{D} .
- 5) **Model assisted evolution (lines 8-11):** In the local iteration, the population \mathcal{P} will be improved by generation and selection operator based on current surrogate model. More details will be presented in next section.
- 6) **Select promising solution (line 12):** According to the multi-acquisition function strategy, the best promising solution will be selected and considered as the offspring solution.
- 7) **Update dataset \mathcal{D} (lines 13-14):** The offspring solution are evaluated by real objective function and merged into \mathcal{A} .

The major components in SA-EDA are introduced in detail in the following section.

B. Generate Solutions

This section will introduce how to generate new solutions. The (VWH) model and evaluated-free local search use global statistical information and location information of visited solutions, respectively.

1) *Variable-width Histogram Model:* A VWH [21] is introduced to improve the capture of high-precision optimal solutions. The VWH model mainly focuses on promising areas. The other areas have a very low probability of avoiding premature convergence, which is critical for expensive optimization under limited fitness evaluations.

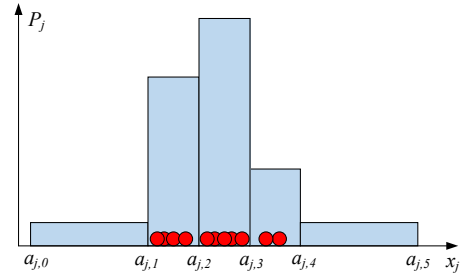


Fig. 1: Illustration of VWH with continuous decision variables with $H=5$.

The joint probability distribution of a VWH can be formulated as a product of univariate marginal distribution in equation (4), where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ denotes a random variable vector and $P_j(x_j)$ is the marginal probability function of x_j . Let H represent the number of bins. As show in Fig. 1, The h th bin can be described by a tuple $\langle a_{j,h-1}, a_{j,h}, P_{j,h} \rangle$. The $a_{j,h-1}$ and $a_{j,h}$ are the boundaries of the bin and $P_{j,h}$ denotes the probability that x_j is from the interval $[a_{j,h-1}, a_{j,h})$. In sampling $P_j(x_j)$, we selecting a bin according to the probability $P_{j,h}$ and uniformly sampling a component x_j from $[a_{j,h-1}, a_{j,h})$.

The major characteristic of VWH is the adaptive univariate model and specific details are as follows. The j th variable x_j

is divided into H bins $[a_{j,h}, a_{j,h+1})_{j=0}^{H-2}$ and $[a_{j,H-1}, a_{j,H}]$, where $a_{j,0}$ and $a_{j,H}$ are equivalent to the boundaries search space of $[a_j, b_j]$. Set

$$\begin{aligned} a_{j,1} &= \max \{x_{j,\min}^1 - 0.5(x_{j,\min}^2 - x_{j,\min}^1), a_j\} \\ a_{j,M-1} &= \min \{x_{j,\max}^1 + 0.5(x_{j,\max}^1 - x_{j,\max}^2), b_j\} \end{aligned} \quad (7)$$

where $x_{j,\min}^1$ and $x_{j,\min}^2$ are the first and second minimum values, respectively, and $x_{j,\max}^1$ and $x_{j,\max}^2$ are the first and second maximum values, respectively. Other $M-2$ bins have the same width as follow

$$a_{j,h} - a_{j,h-1} = \frac{1}{H} (a_{j,H-1} - a_{j,1}) \quad (8)$$

This division aims to guarantee that all solutions fall into the middle $M-2$ bins. Let $C_{j,h}$ denote the count of solutions in the h th bin for x_j . For each bin to have a probability for selection in sampling, the rules of calculating $C_{i,m}$ is defined as

$$C_{j,h} = \begin{cases} C_{j,h} + 1 & \text{if } 1 < h < H \\ 0.1 & \text{if } h = 1, H, \text{ and } a_{j,h} > a_{j,h-1} \\ 0 & \text{if } h = 1, H, \text{ and } a_{j,h} = a_{j,h-1}. \end{cases} \quad (9)$$

The probability $P_{j,h}$ could be calculating as

$$P_{j,h} = \frac{C_{j,h}}{\sum_{j=1}^H C_{i,h}} \quad (10)$$

After this definition, the probability $P_{j,h}$ of bins in promising area is higher than the end two bins which is consider the EDA model focus on the promising area to enhance the convergence ability. When we sampling a \mathbf{x} from the $P(X)$, randomly select a bin h according the probability $P_{j,h}$, $h = 1, \dots, H$ for each dimension j of n . Then uniformly pick a value x_j from $[a_{j,h-1}, a_{j,h})$ when $h < H$ or $[a_{i,H-1}, a_{j,H}]$ when $h = H$. Finally, a offspring solution $\mathbf{x} = x_1, x_2, \dots, x_n$ is sampled. The H in this work is set to 15.

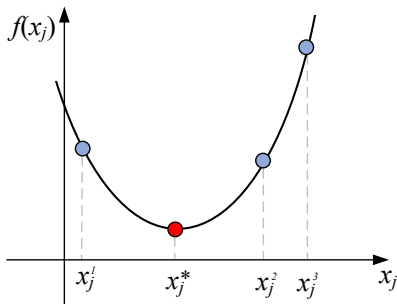


Fig. 2: Illustration of local quadratic approximation to detect local optimal point.

2) *Evaluated-free Local Search*: The EDA model is a global model that learns the distribution information of populations. An evaluated-free local search [21] is introduced in this section to improve the exploration of local areas. The landscape of each decision variable x_j can be approximated by a quadratic model as:

$$f = ax_j^2 + bx_j + c \quad (11)$$

if $a > 0$, the optimal solution in j dimension is calculated as $x_j^* = -\frac{b}{2a}$. the parameters in equation (11) can be calculated with three known points $\{(x_j^k, f(x_j^k))\}_{k=1}^3$.

$$\begin{aligned} a &= \frac{1}{x_j^2 - x_j^3} \left[\frac{f(x_j^1) - f(x_j^2)}{x_j^1 - x_j^2} - \frac{f(x_j^1) - f(x_j^3)}{x_j^1 - x_j^3} \right] \\ b &= \frac{f(x_j^1) - f(x_j^2)}{x_j^1 - x_j^2} - a(x_j^1 + x_j^2). \end{aligned} \quad (12)$$

By this method, a solutions \mathbf{x} will be improved by a simple data set $\{(\mathbf{x}^k, f(\mathbf{x}^k))\}_{k=1}^3$.

Algorithm 2: Generation_solution

Input : \mathcal{P} (parent solutions);

- 1 $\mathcal{P} \leftarrow \text{sort}(\mathcal{P}, \text{'ascend'})$.
- 2 $\mathcal{Q} \leftarrow \emptyset$.
- 3 $P(X) \leftarrow \text{EDA}(\mathcal{P})$. /* build EDA model */
- 4 **for** $i \leftarrow 1$ **to** N **do**
- 5 $\mathbf{u}^i \leftarrow \text{sample}(P(X))$.
- 6 $k \leftarrow \text{random_select}\{1, 2, \dots, \lfloor P_b N \rfloor - 2\}$.
- 7 /* select a parent */
- 8 **for** $j \leftarrow 1$ **to** n **do**
- 9 **if** $\text{rand}() < P_c$ **then**
- 10 $u_j^i \leftarrow \text{LS}(\{x_j^i, f(x_j^i)\}, i = k, k+1, k+2)$.
- 11 /* local search */
- 12 **end**
- 13 $u_j^i \leftarrow \begin{cases} 0.5(x_j^i + a_j) & \text{if } u_j^i < a_j \\ u_j^i & \text{if } a_j \leq u_j^i \leq b_j \\ 0.5(x_j^i + b_j) & \text{if } u_j^i > b_j \end{cases}$.
- 14 /* check boundary */
- 15 **end**
- 16 $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathbf{u}^i\}$.
- 17 **end**

Output: \mathcal{Q} (trial solutions).

3) *Reproduction*: The trial solutions are obtained by Algorithm 2. In line1, the populations are sorted by their objective by real function evaluation or surrogate approximation. Next, build the EDA model according to the equation (10). Then, a new solution \mathbf{u}^i is sampled from EDA model (line5). To generate high-quality solutions, $\lfloor P_b \cdot N \rfloor$ best solutions are used to guild local search. P_c controls the contribution of the local search. Lines 6-10 in Algorithm 2 shows the details. In line11, the solution value will be checked by the lower and upper boundaries (a_i, b_i) of the black-box function. Finally, iterate N times to produce a population \mathcal{Q} . In this paper, the value of P_b and P_c are both set to 0.2.

C. Multi-acquisition Criteria

Gaussian process is used as the surrogate model for objective value prediction. At the same time, various can be given. The acquisition functions can guide the selection to balance

the search behavior of exploration and exploitation. Based on this, common acquisition functions are used as model management strategies, e.g., expected improvement (EI), probability improvement (PI), and lower confidence bound (LCB). The definitions are as follows.

$$ei(\mathbf{x}) = (\mu(\mathbf{x}) - \mathbf{x}^*)\Phi\left(\frac{\mu(\mathbf{x}) - \mathbf{x}^*}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\Phi\left(\frac{\mu(\mathbf{x}) - \mathbf{x}^*}{\sigma(\mathbf{x})}\right) \quad (13)$$

$$pi(\mathbf{x}) = \Phi\left(\frac{\mathbf{x}^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) \quad (14)$$

$$lcb(\mathbf{x}) = -(\mu(\mathbf{x}) - \sqrt{\beta}\sigma(\mathbf{x})) \quad (15)$$

where $\mu(\mathbf{x})$, also know as $\hat{f}(\mathbf{x})$, is the predict value of testing point \mathbf{x} and $\sigma(\mathbf{x})$ is various.

Considering that different acquisitions led to conflicting results, we introduce a hedge strategy [23] to ensure the robustness of the results. The details are in Algorithm 3. Two situations were proposed for surrogate assisted selection.

- ‘default’: Only the predicted value μ of surrogate for f is used when the convergence is considered (line 7).
- ‘hedge’: Three acquisitions are introduced to enhance robustness in balancing exploration and exploitation (lines 3-5). Three solutions are selected according to acquisitions independently. Probabilistically choose one of the above these at each generation. The probability is calculated by $\text{softmax}(g_i)$. The g_i is the cumulative reward that is updated in each generation, and details can be found in [23].

Algorithm 3: SA_selection

Input : \mathcal{P} (population);
 \mathcal{M} (surrogate model);
 opt (options of method);

```

1  $\mu, \sigma \leftarrow \text{predict}(\mathcal{M}, \mathcal{P})$ 
2 if  $opt$  is ‘hedge’ then
3    $ei, pi, lcu \leftarrow$  calculate three kinds of acquisitions
   base on  $\mu, \sigma$  according to equations (13), (14),
   (15).
4    $offs \leftarrow$  select three solutions with the smallest
   value of  $-ei, -pi, lcu$  from  $\mathcal{P}$ .
5    $offs \leftarrow \mathbf{GP-hedge}(offs)$ 
6 else
7    $offs \leftarrow$  select a solution with the smallest  $\mu$  from
    $\mathcal{P}$ .
8 end
```

Output: $offs$ (offspring solutions).

In this work, the ‘default’ situation is used to select solutions in local iteration to make the solution converge quickly to the optimal region of the current model. The ‘hedge’ situation maintains the diversity of the solution to evaluation and reduces model uncertainty.

IV. EXPERIMENTAL RESULTS

This section studies the performance of the proposed SA-EDA approach. We first provide the experimental settings and comparison methods. Then, we compare the performance in small-scale problems. Next, performance on the medium-scale problem is discussed. Finally, a rover trajectories optimization is used to study the proposed method of solving a complex problem.

A. Experimental settings

1) *Test suit:* Two test suits, LZG [9] and YLL [24], are employed for empirical study. The LZG test suit contains four test functions: Ellipsoid, Rosenbrock, Ackley, and Griewank with unimodal, gully, and multimodal landscapes. In the YLL test suite, $f1-f4$ are unimodal, $f5, f8-f13$ are multimodal, $f6$ is step and $f7$ is with random noise. The dimension n of the problems is set to 20 for small-scale and 50 for medium-scale.

2) *Algorithms in Study:* Seven algorithms, i.e. CMA-ES [25], EDALS [21], FCPS [13], SAMSO [26], SACOSO [27], Skopt¹, and GPEME [9] are chosen for study. They can be divided into three categories:

- EDAs: CMA-ES is a general EDA, and EDALS is a univariate histogram marginal model based EDA with a global surrogate model. They are not specially designed for expensive optimization.
- Bayesian optimization: Skopt is an efficient global optimization algorithm in the BO framework, which applies GP as the surrogate model.
- Surrogate assisted EAs: FCPS uses a fuzzy K nearest neighbor-based classification model to evaluate candidate solutions. GPEME uses GP models to evaluate candidate solutions. SAMSO and SACOSO are surrogate assisted PSO algorithms with RBF.

3) *Parameter Settings:* In the empirical study, we use the recommended parameters in the original literature for each algorithm for a fair comparison², and the details are given below.

- Termination condition: The maximum number of FEs is adopted as the termination condition, set at 500 for all instances.
- Population size: For CMA-ES, EDALS and FCPS, SAMSO the N is set to 30. Set $N = 50$ for GPEME and SA-EDA.
- Parameters of SA-EDA: t_{max}, α are set to 5, 50 respectively.
- Parameters of compared algorithms: Default setting according to original version.

Each algorithm is executed on each test instance for 30 independent runs to overcome randomness. The Wilcoxon rank

¹<https://github.com/scikit-optimize/scikit-optimize>

²CMA-ES, SAMSO, and SACOSO are implemented in Platemo [28]
 Skopt:<https://github.com/scikit-optimize/scikit-optimize>
 FCPS, EDALS, and GPEME are implemented by us based on the original report.

TABLE I: Statistics of median results obtained by eight comparison algorithms on LZG and YLL test suites with $n = 20$.

problem	CMAES	EDALS	Skopt	FCPS	SAMSO	SACOSO	GPEME	SA-EDA
Ellipsoid	1.50e+02[8](−) (4.25e+01)	7.17e+01[6](−) (1.56e+01)	1.81e+00[3](−) (2.29e+00)	1.30e+02[7](−) (3.13e+01)	2.41e+00[4](−) (5.93e+00)	5.48e+00[5](−) (1.00e+01)	3.20e-01[2](−) (1.77e-01)	1.02e-02[1] (3.55e-03)
Rosenbrock	3.03e+02[6](−) (7.50e+01)	2.37e+02[5](−) (4.02e+01)	6.45e+02[8](−) (1.62e+03)	3.22e+02[7](−) (1.05e+02)	2.50e+01[1](+) (7.78e+00)	6.10e+01[3](≈) (2.67e+01)	1.27e+02[4](−) (4.32e+01)	6.01e+01[2] (2.49e+01)
Ackley	1.59e+01[7](−) (1.28e+00)	1.33e+01[4](−) (7.37e-01)	1.49e+01[6](−) (6.91e+00)	1.48e+01[5](−) (1.00e+00)	1.90e+01[8](−) (7.24e-01)	4.14e+00[3](−) (1.11e+00)	3.77e+00[2](−) (5.52e-01)	2.70e+00[1] (3.79e-01)
Griewank	5.09e+01[7](−) (1.06e+01)	2.96e+01[6](−) (7.62e+00)	1.19e+00[3](−) (3.04e-01)	5.46e+01[8](−) (1.33e+01)	2.68e+01[5](−) (1.75e+01)	2.41e+00[4](−) (9.99e-01)	1.17e+00[2](−) (8.76e-02)	9.85e-01[1] (2.90e-02)
YLLF01	6.32e+03[8](−) (1.98e+03)	3.17e+03[6](−) (5.92e+02)	2.07e+01[3](−) (5.61e+01)	5.37e+03[7](−) (1.86e+03)	5.31e+02[5](−) (3.54e+02)	1.91e+02[4](−) (1.95e+02)	1.67e+01[2](−) (1.09e+01)	1.36e+00[1] (6.55e-01)
YLLF02	1.15e+02[6](−) (2.20e+02)	2.67e+01[3](+) (3.93e+00)	8.08e+16[8](−) (2.18e+17)	2.41e+01[2](+) (3.67e+00)	3.02e+01[4](+) (1.48e+01)	2.27e+01[1](+) (1.58e+01)	5.03e+02[7](−) (1.04e+03)	4.26e+01[5] (8.69e+00)
YLLF03	2.09e+04[4](≈) (5.68e+03)	2.32e+04[6](≈) (4.21e+03)	5.29e+04[8](−) (3.37e+04)	1.26e+04[1](+) (3.48e+03)	1.60e+04[2](+) (1.36e+04)	1.84e+04[3](≈) (9.78e+03)	2.99e+04[7](−) (6.03e+03)	2.16e+04[5] (5.12e+03)
YLLF04	4.03e+01[5](−) (1.25e+01)	3.28e+01[4](−) (3.26e+00)	7.86e+01[8](−) (3.14e+01)	4.17e+01[6](−) (6.55e+00)	7.28e+01[7](−) (5.50e+00)	1.72e+01[1](+) (5.84e+00)	3.09e+01[3](−) (8.94e+00)	2.19e+01[2] (5.25e+00)
YLLF05	3.11e+06[7](−) (1.43e+06)	1.13e+06[5](−) (5.76e+05)	1.25e+06[6](−) (1.27e+06)	4.71e+06[8](−) (2.84e+06)	1.38e+05[3](≈) (1.28e+05)	2.98e+04[1](+) (7.29e+04)	3.05e+05[4](−) (1.60e+05)	8.39e+04[2] (3.77e+04)
YLLF06	5.79e+03[7](−) (1.20e+03)	3.20e+03[6](−) (5.04e+02)	2.11e+01[2](≈) (5.38e+01)	6.26e+03[8](−) (1.28e+03)	7.67e+02[5](−) (5.09e+02)	1.54e+02[4](−) (1.06e+02)	2.32e+01[3](−) (1.06e+01)	3.00e+00[1] (1.37e+00)
YLLF07	1.62e+00[6](−) (6.91e-01)	6.67e-01[5](−) (2.75e-01)	3.31e+00[8](−) (8.29e+00)	2.09e+00[7](−) (9.17e-01)	4.36e-01[4](−) (2.47e-01)	1.60e-01[1](≈) (1.22e-01)	2.90e-01[3](−) (9.39e-02)	1.82e-01[2] (6.60e-02)
YLLF08	5.71e+03[6](−) (3.16e+02)	4.66e+03[2](−) (3.32e+02)	7.45e+03[8](−) (1.05e+03)	5.52e+03[4](−) (2.88e+02)	5.54e+03[5](−) (3.28e+02)	5.85e+03[7](−) (3.57e+02)	4.70e+03[3](−) (7.05e+02)	2.95e+03[1] (6.06e+02)
YLLF09	1.84e+02[7](−) (2.29e+01)	1.57e+02[6](−) (1.31e+01)	2.56e+02[8](−) (5.09e+01)	1.02e+02[2](+) (2.09e+01)	1.05e+02[3](≈) (5.67e+01)	7.06e+01[1](+) (2.61e+01)	1.34e+02[5](−) (2.06e+01)	1.22e+02[4] (1.92e+01)
YLLF10	1.61e+01[7](−) (1.90e+00)	1.33e+01[5](−) (9.54e-01)	5.36e+00[4](−) (4.86e+00)	1.45e+01[6](−) (1.34e+00)	1.98e+01[8](−) (5.36e-01)	3.97e+00[3](−) (9.19e-01)	1.23e+00[2](−) (5.35e-01)	2.41e-01[1] (9.00e-02)
YLLF11	5.48e+01[8](−) (1.16e+01)	3.14e+01[6](−) (7.17e+00)	1.07e+00[2](−) (6.58e-02)	4.90e+01[7](−) (1.24e+01)	2.98e+01[5](−) (2.10e+01)	2.18e+00[4](−) (9.42e-01)	1.17e+00[3](−) (9.31e-02)	9.75e-01[1] (3.59e-02)
YLLF12	6.01e+05[6](−) (7.89e+05)	6.09e+04[4](−) (8.76e+04)	1.13e+06[7](−) (9.82e+05)	2.46e+06[8](−) (3.04e+06)	2.46e+01[2](+) (3.71e+01)	2.57e+00[1](+) (1.73e+00)	4.29e+05[5](−) (5.69e+05)	1.81e+04[3] (6.20e+04)
YLLF13	5.47e+06[4](+) (2.96e+06)	1.10e+06[3](+) (5.84e+05)	1.14e+12[8](−) (3.99e+12)	1.08e+07[5](+) (8.07e+06)	1.75e+04[2](+) (2.62e+04)	4.67e+01[1](+) (1.32e+02)	4.76e+10[7](−) (3.08e+10)	1.36e+10[6] (7.24e+09)
mean rank	6.41	4.82	5.88	5.76	4.29	2.76	3.76	2.29
+ / − / ≈	1/15/1	2/14/1	0/16/1	4/13/0	5/10/2	6/8/3	0/17/0	0/0/0

sum test [29] is used to compare the experimental results, where ‘+’, ‘−’, and ‘≈’ in the tables indicate the value obtained by an algorithm is smaller than, greater than, or similar to that obtained by the RCPS based version at a 95% significance level. The value in the brackets denotes the corresponding rank.

B. Results on small-scale problem

For small-scale problems, Table I lists the statistical results of 8 algorithms. The value in the brackets denotes the corresponding ranks, and the top one rank is marked with grey shadows on each line, respectively. The last lines of the table show the mean ranks and Wilcoxon rank sum test of each algorithm on all instances. The table shows that SA-EDA achieves the best mean rank of 2.29 among the 8 algorithms. Skopt achieves the 5.88 mean rank. Of the four SAEAs, SACOSO achieved the best results, and FCPS has poor performance. CMA-ES, the only no-surrogate method, achieved the worst mean rank of 6.41. Even though EDALS uses surrogate models, there are still many fitness evaluations in each generation, so it achieved a 4.82 mean rank. Similar

conclusions can be drawn from the Wilcoxon rank sum test result. From all instance, SA-EDA obtained 15, 14, 16, 13, 10, 8, 17 better, 1, 2, 0, 4, 5, 6, 0 worse and 1, 1, 1, 0, 2, 3, 0 similar optimal solution than CMA-ES, EDALS, Skopt, FCPS, SAMSO, SACOSO, GPEME. The median runtime performance curves are shown in Fig 3. The results are consistent with Table I. SA-EDA shows the best convergence performances. Especially on YLLF08 and YLLF10, SA-EDA shows outstanding advantages.

C. Results on medium-scale problem

This subsection studies the performance of SA-EDA in 50 dimensional problems. Due to the dimensional catastrophe, Skopt and GPEME are not involved in the comparison. Three comparison algorithms, CMAES, FCPS, and SAMSO, represent the model-free, classification model, and regression model based algorithms respectively. The statistical results are shown in Table II of four algorithms with $n = 50$. SA-EDA achieves a best mean rank of 1.41, which is better than the SAMSO, a specifically designed method for the medium-scale problem.

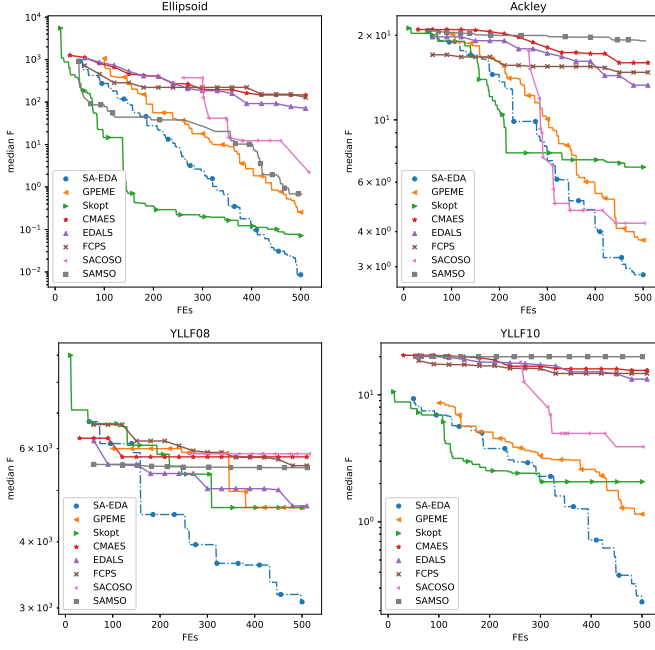


Fig. 3: The median runtime performance of 6 algorithms on LZG and YLL test suit.

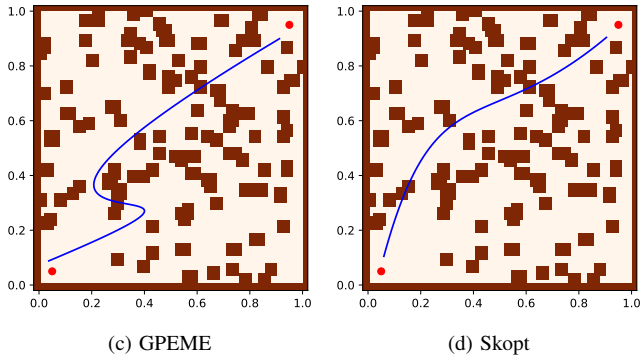
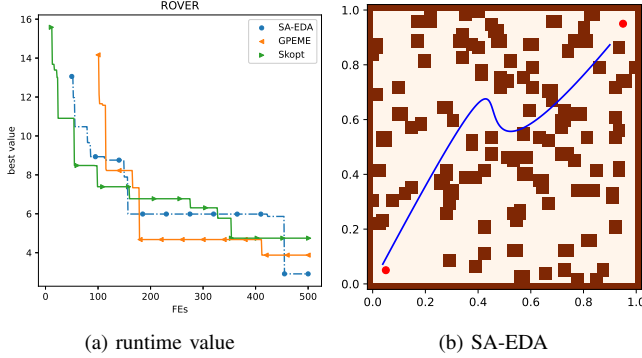


Fig. 4: The optimization result of ROVER. (a) The best objective value versus fitness evaluation times by three algorithms. (b,c,d) Best example trajectories found by SA-EDA, GPEME and Skopt.

The Wilcoxon rank sum test also shows that the SA-EDA works better on most instances than other algorithms.

TABLE II: Statistics of median results obtained by four comparison algorithms on LZG and YLL test suites with $n = 50$.

problem	CMAES	FCPS	SAMSO	SA-EDA
Ellipsoid	1.91e+03[4](-) (2.77e+02)	1.61e+03[3](-) (3.39e+02)	1.01e+03[2](≈) (1.29e+03)	8.08e+02[1] (1.72e+02)
Rosenbrock	2.09e+03[4](-) (4.12e+02)	1.90e+03[3](-) (5.11e+02)	1.20e+03[1](≈) (1.32e+03)	1.33e+03[2] (2.86e+02)
Ackley	1.85e+01[3](-) (9.68e-01)	1.75e+01[2](-) (6.00e-01)	1.90e+01[4](-) (9.21e-01)	1.56e+01[1] (6.55e-01)
Griewank	2.38e+02[2](-) (3.80e+01)	2.69e+02[3](-) (6.72e+01)	5.78e+02[4](-) (3.57e+02)	1.73e+02[1] (2.23e+02)
YLLF01	2.61e+04[2](-) (3.48e+03)	2.95e+04[3](-) (5.86e+03)	7.22e+04[4](-) (3.19e+04)	1.44e+04[1] (2.89e+03)
YLLF02	8.62e+13[3](-) (3.40e+14)	8.92e+01[1](+) (8.24e+00)	1.35e+18[4](-) (3.61e+18)	1.64e+07[2] (2.33e+01)
YLLF03	1.36e+05[3](≈) (2.57e+04)	8.24e+04[1](+) (1.68e+04)	2.78e+05[4](-) (8.57e+04)	1.36e+05[2] (2.49e+04)
YLLF04	9.16e+01[4](-) (1.26e+01)	5.88e+01[2](≈) (5.58e+00)	8.99e+01[3](-) (2.91e+00)	5.85e+01[1] (2.72e+00)
YLLF05	3.78e+07[2](-) (1.25e+07)	4.49e+07[3](-) (1.82e+07)	2.21e+08[4](-) (9.88e+07)	1.52e+07[1] (4.04e+06)
YLLF06	2.82e+04[2](-) (5.82e+03)	2.92e+04[3](-) (7.35e+03)	8.75e+04[4](-) (2.39e+04)	1.50e+04[1] (3.03e+03)
YLLF07	3.14e+01[3](-) (9.15e+00)	2.96e+01[2](-) (1.28e+01)	1.59e+02[4](-) (8.25e+01)	1.49e+01[1] (3.87e+00)
YLLF08	1.65e+04[4](-) (5.98e+02)	1.63e+04[2](-) (6.11e+02)	1.64e+04[3](-) (9.43e+02)	1.38e+04[1] (8.40e+02)
YLLF09	6.21e+02[4](-) (4.52e+01)	3.80e+02[1](+) (3.15e+01)	4.98e+02[3](≈) (1.13e+02)	4.87e+02[2] (2.41e+01)
YLLF10	1.89e+01[3](-) (1.45e+00)	1.72e+01[2](-) (8.67e-01)	1.98e+01[4](-) (5.83e-01)	5.25e+00[1] (2.93e-01)
YLLF11	2.47e+02[2](-) (3.82e+01)	2.52e+02[3](-) (6.26e+01)	5.28e+02[4](-) (3.47e+02)	1.61e+02[1] (3.08e+01)
YLLF12	4.07e+07[2](-) (2.32e+07)	5.44e+07[3](-) (3.49e+07)	4.82e+08[4](-) (2.65e+08)	1.17e+07[1] (4.77e+06)
YLLF13	1.03e+08[1](+) (3.96e+07)	1.58e+08[2](+) (1.01e+08)	7.61e+08[3](+) (4.43e+08)	2.03e+12[4] (7.69e+11)
mean rank	2.82	2.29	3.47	1.41
+ / - / ≈	1/15/1	4/12/1	1/13/3	0/0/0

D. Result on Rover Trajectories Optimizing

This section shows the results of solving the rover trajectories optimizing problem. The two-dimensional positions of the 30 control points will be optimized to generate a trajectory that minimizes collision loss. Details description of the problem can be found in [30]. In brief, this is a 60n complex optimization problem accompanied by a non-smooth, discontinuous, and concave objective function. Two kinds of algorithms, GPEME and Skopt, are involved compared to SA-EDA. The experiment was run 10 times independently with each algorithm, and Fig 4a plots the best run time performance. The figure suggests that SA-EDA achieves the best result within three algorithms. Fig 4b-4d shows the final obtained trajectory in 2-D space. Compared with the other two results, the trajectory obtained by SA-EDA avoids most obstacles from the start point to the goal point. The results show that SA-EDA can handle complex problems with limited evaluation times.

V. CONCLUSION

This paper proposed a surrogate assisted estimation of distribution algorithm (SA-EDA) with the multi-acquisition function strategy. An efficient variable-width histogram is introduced into SAEAs to improve the convergence ability of the population by focusing on potential regions. The evaluated-free quadratic model is used for local search to enhance the exploration of the local area. Offspring selection is guided by a multi-acquisition function strategy to overcome conflict and instability. SA-EDA has been compared with 7 state-of-the-art algorithms on 17 widely used test instances from LZG and YLL test suites, which contain a Bayesian optimization algorithm, four surrogate assisted algorithms, and two eliminate distribution algorithms. The statistical results show that SA-EDA gets good results on small- and medium-scale problems in some test instances. Moreover, SA-EDA has an edge in complex trajectories optimization problems.

There are several research avenues worthwhile exploring in the future. First, some other EDAs should be investigated in SAEAs. Secondly, the VWH with local search has efficient search capabilities that can be used in other challenge optimization problems. Finally, adaptive multi-acquisitions should be investigated to enrich the model management study.

REFERENCES

- [1] D. Ding, J. Xia, L. Yang, and X. Ding, "Multiobjective Optimization Design for Electrically Large Coverage: Fragment-Type Near-Field/Far-Field UHF RFID Reader Antenna Design," *IEEE Antennas and Propagation Magazine*, vol. 60, no. 1, pp. 27–37, 2018-02.
- [2] Z. Lu, I. Whalen, V. Boddetti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, "NSGA-Net: Neural architecture search using multi-objective genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2019-07-13, pp. 419–427.
- [3] Z. Shangquan, L. Lin, W. Wu, and B. Xu, "Neural process for black-box model optimization under bayesian framework," in *Proceedings of the AAAI 2021 Spring Symposium on Combining Artificial Intelligence and Machine Learning with Physical Sciences, Stanford, CA, USA, March 22nd - to - 24th, 2021*, vol. 2964. CEUR-WS.org, 2021.
- [4] P. I. Frazier, "A tutorial on bayesian optimization," *CoRR*, vol. abs/1807.02811, 2018. [Online]. Available: <http://arxiv.org/abs/1807.02811>
- [5] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011-06-01.
- [6] Y. Lian and M.-S. Liou, "Multiobjective optimization using coupled response surface model and evolutionary algorithm," *AIAA journal*, vol. 43, no. 6, pp. 1316–1325, 2005.
- [7] A. I. Khuri and S. Mukhopadhyay, "Response surface methodology," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 2, pp. 128–149, 2010.
- [8] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, 2017.
- [9] B. Liu, Q. Zhang, and G. G. E. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, 2014.
- [10] L. Willmes, T. Back, Y. Jin, and B. Sendhoff, "Comparing neural networks and kriging for fitness approximation in evolutionary optimization," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 1. IEEE, 2003, pp. 663–670.
- [11] H.-r. Li, F.-z. He, and X.-h. Yan, "Ibea-svm: an indicator-based evolutionary algorithm based on pre-selection with classification guided by svm," *Applied Mathematics-A Journal of Chinese Universities*, vol. 34, no. 1, pp. 1–26, 2019.
- [12] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, 2019.
- [13] A. Zhou, J. Zhang, J. Sun, and G. Zhang, "Fuzzy-classification assisted solution preselection in evolutionary optimization," in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, pp. 2403–2410.
- [14] S. D. Handoko, C. K. Kwok, and Y.-S. Ong, "Feasibility Structure Modeling: An Effective Chaperone for Constrained Memetic Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 740–758, 2010-10.
- [15] S. Gelly, S. Ruetz, and O. Teytaud, "Comparison-Based Algorithms Are Robust and Randomized Algorithms Are Anytime," *Evolutionary Computation*, vol. 15, no. 4, pp. 411–434, 2007-12.
- [16] T. P. Runarsson, "Ordinal Regression in Evolutionary Computation," in *Parallel Problem Solving from Nature - PPSN IX*, ser. Lecture Notes in Computer Science, T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds. Springer Berlin Heidelberg, 2006, vol. 4193, pp. 1048–1057.
- [17] I. Loshchilov, M. Schoenauer, and M. Sebag, "Comparison-based optimizers need comparison-based surrogates," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2010, pp. 364–373.
- [18] A. Dushatskiy, A. M. Mendrik, T. Alderliesten, and P. A. N. Bosman, "Convolutional neural network surrogate-assisted GOMEA," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO)*. ACM, 2019-07-13, pp. 753–761.
- [19] H. Hao, J. Zhang, X. Lu, and A. Zhou, "Binary Relation Learning and Classifying for Preselection in Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1125–1139, 2020-12.
- [20] C. E. Rasmussen and C. K. Williams, *Gaussian Process for Machine Learning*. MIT Press Cambridge, MA, 2006.
- [21] A. Zhou, J. Sun, and Q. Zhang, "An estimation of distribution algorithm with cheap and expensive local search methods," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 807–822, 2015.
- [22] M. D. McKay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000-02.
- [23] M. Hoffman, E. Brochu, and N. de Freitas, "Portfolio allocation for bayesian optimization," in *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, F. G. Cozman and A. Pfeffer, Eds. AUAI Press, 2011, pp. 327–336.
- [24] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [25] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [26] F. Li, X. Cai, L. Gao, and W. Shen, "A surrogate-assisted multiswarm optimization algorithm for high-dimensional computationally expensive problems," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1390–1402, 2021.
- [27] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, 2017.
- [28] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017-11.
- [29] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. John Wiley & Sons, 2013, vol. 751.
- [30] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka, "Batched large-scale bayesian optimization in high-dimensional spaces," in *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, ser. Proceedings of Machine Learning Research, A. J. Storkey and F. Pérez-Cruz, Eds., vol. 84. PMLR, 2018, pp. 745–754.