

---

# MULTI-SURROGATE ASSISTED EFFICIENT GLOBAL OPTIMIZATION FOR DISCRETE PROBLEMS

---

Qi Huang\*, Roy de Winter, Bas van Stein, Thomas Bäck, Anna V. Kononova

Leiden Institute of Advanced Computer Science

Leiden University

Leiden, Netherlands

{q.huang, r.de.winter, b.van.stein, t.h.w.baeck, a.kononova}@liacs.leidenuniv.nl

## ABSTRACT

Decades of progress in simulation-based surrogate-assisted optimization and unprecedented growth in computational power have enabled researchers and practitioners to optimize previously intractable complex engineering problems. This paper investigates the possible benefit of a concurrent utilization of multiple simulation-based surrogate models to solve complex discrete optimization problems. To fulfill this, the so-called Self-Adaptive Multi-surrogate Assisted Efficient Global Optimization algorithm (SAMA-DiEGO), which features a two-stage online model management strategy, is proposed and further benchmarked on fifteen binary-encoded combinatorial and fifteen ordinal problems against several state-of-the-art non-surrogate or single surrogate assisted optimization algorithms. Our findings indicate that SAMA-DiEGO can rapidly converge to better solutions on a majority of the test problems which shows the feasibility and advantage of using multiple surrogate models in optimizing discrete problems.

**Keywords** Discrete Optimization · Surrogate Assisted Optimization · Bayesian Optimization · Global Optimization

## 1 Introduction

Due to the increase in computational power, it becomes feasible in many engineering fields to optimize new or existing components to ever-increasing demands in terms of sustainability, efficiency, and safety. These challenges and arising expectations to tackle them, give space to the development and application of mathematical optimization, which comprises a series of techniques and can be found nearly everywhere in modern-day engineering fields, e.g., aircraft/vehicle design [14] and chemical engineering [24]. Nowadays, instead of manually finding the best solutions or designs, a computer with its enormous computational power can tirelessly and smartly go through the search space of these models under certain guidelines called optimization algorithms.

The primary goal of mathematical optimization is to find a desirable (optimal) solution(s) from a set of candidate solutions based on their performance on objective (target) function(s). Assuming the outputs of objective functions are accessible within reasonable computation time, a group of well-established solvers is the query-based solvers, e.g., evolutionary algorithms and the covariance matrix adaptation evolution strategies. However, in real-world applications, there are cases where the objective functions are expensive to evaluate. Consequently, it is infeasible to allow a solver to evaluate thousands of (feasible) solutions using expensive objective functions due to the unacceptable need for execution time and resources. To tackle expensive optimization problems, Jones et al. [23] proposed the well-known Efficient Global Optimization (EGO), a.k.a, Bayesian Optimization (BO) algorithm, which efficiently solves expensive optimization problems by constructing a so-called surrogate model (a.k.a., meta-model or meta-heuristic) to simulate the objective function and shifting the optimization process that was once based on an expensive objective function to a cheaper surrogate model.

Decades of development have resulted in a wide variety of surrogate-assisted algorithms, each with distinct characteristics and suitable application scenarios. Bhosekar and Ierapetritou [6] made an introductory review in 2018, which

---

\*The corresponding author

summarized frequently used surrogate models; famous derivative-free optimization algorithms (solvers) that are suitable to combine with surrogates; as well as accompanied sampling strategies and validation metrics for surrogate models. However, the majority of existing surrogate-assisted/based optimization reviews and works are primarily focused on handling (expensive) optimization problems defined in continuous search spaces. This type of search space is exclusively comprised of numerical variables, each of which has an infinite number of values between any two values. Obviously, there exist real-world optimization problems, e.g., vehicle routing and scheduling [30], that is entirely built upon a discrete search space, where each variable has a finite number of choices.

In contrast to the flourished development of EGO as well as online surrogate management for real-valued optimization, Jin [22] pointed out the scarcity of studies on surrogate-based/assisted discrete/combinatorial optimization in 2011. Later in 2017, a survey by Bartz-Beielstein and Zaefferer [2] identified the key issue as finding suitable surrogate models for discrete problems. This challenge becomes critical if EGO algorithms are applied, as these solvers normally rely on a single surrogate model predefined beforehand. Therefore, it is a natural choice to consider the possibility of concurrently using multiple surrogate models, which are all capable of handling discrete parameter spaces, to tackle complex discrete problems. Hence, this research focuses on the feasibility and potential benefit of utilizing multiple surrogate models in an EGO-styled algorithm to solve purely discrete problems. To the best of our knowledge, there is a lack of noteworthy studies on this topic.

Given this motivation, a novel multiple-surrogate-assisted optimization algorithm for discrete optimization problems is proposed in this manuscript. The performance of the algorithm is evaluated on both benchmark combinatorial, and integer-valued problems. The algorithm is based on the well-established Efficient Global Optimization (EGO) [23] and features a two-stage online model management strategy inspired by [1]. To address the term *multiple*, four types and a total of thirty-one surrogate models that universally accommodate discrete problems are implemented. Additionally, the source codes and supplementary materials can be found in [20].

The remainder of the paper is organized as follows: In Section 2, the relevant literature on online management of multiple surrogate models and feasible surrogate models for discrete optimization are provided. The proposed multi-surrogate-assisted algorithm is presented and block-by-block introduced in Section 3. Next, the algorithms are benchmarked with the settings outlined in Section 4, and the experimental results are discussed in Section 5. Lastly, the conclusions and suggestions for future work is given in Section 6.

## 2 Related Literature

### 2.1 Online Management of Multiple Surrogate Models

Utilizing multiple surrogate models in optimization is a well-practiced idea. Online management of multiple surrogate models denotes the process of maintaining and determining the most suitable proxy (surrogate model) for an objective function while doing optimization. In contrast to offline model management, online management is capable of utilizing incrementally obtained new data samples. Gorissen et al. [15] proposed an Evolutionary Model Selection (EMS) algorithm to determine the best surrogate model type and its hyperparameter by minimizing cross-validation error or Akaike Information Criterion using a modified genetic algorithm. Bagheri et al. [1] selected the best-performed type of radial basis function interpolation on the basis of their median absolute errors obtained on newly acquired data samples (unseen to surrogate models).

Instead of only using the best surrogate out of all candidates, an alternative choice is to create the ensemble (aggregation) of multiple surrogate models through computing a weighted sum of predictions produced by all candidate models trained on existing observations [17] or on heterogeneous features extracted/selected from data samples [16]. Further, it is feasible to adaptively merge a group of surrogate models with regard to fidelity level, i.e., reliability or accuracy of models in approximating the original problem [12].

By all means, the previously described methods aim to select/create a surrogate model out of all candidate models. An alternative strategy is proposed in [35], namely, the Multiple Surrogate Efficient Global Optimization (MSEGO) algorithm. The MSEGO samples new infill points by independently maximizing the expected improvement on all of its surrogate models, and later evaluates all the new infill points samples using the real function. A similar idea is re-visited in [37] and further developed in [3]. In comparison to exclusively using EI in MSEGO, Beaucair et al. [3] suggest employing different infill criteria for different types of surrogate models and therefore concurrently sampling new points from these independent infill criteria. Besides EGO, studies on surrogate-assisted evolutionary algorithms also feature concurrent utilization of multiple surrogates, e.g., Li et al [25] simultaneously used a radial basis function model and a polynomial regression model to assist particle swarm optimization algorithms to converge on expensive continuous problems in the late stage of optimization.

## 2.2 Surrogate Modeling for Discrete Problems

The need for adapting EGO for discrete problems has recently and constantly been emphasized in solving expensive mixed-integer optimization problems. A notable example is hyperparameter tuning and automatic model selection for machine learning tasks, where, the EGO-styled solvers are more commonly referred to as Sequential Model-Based Optimization (SMBO) algorithms [21]. Studying the surrogate models used in SMBO algorithms is a solid entry point for studying the state-of-the-art methodology of applying surrogate models to expensive discrete problems. The application scenario is: given a set of mixed-integer configurations as  $\lambda$  and the corresponding performance of the expensive machine learning model as  $y$ , surrogate models in SMBO are commonly required to model the conditional probability  $p(y|\lambda)$ . With respect to the types of the surrogate model applied in the well-established SMBO algorithms: Sequential Model-based Algorithm Configuration (SMAC) [21] and Mixed-Integer Parallel Efficient Global Optimization (MIP-EGO) [32] use random forest regression; Gaussian process regression (a.k.a Kriging) are utilized in Spearmint optimizer [31] and Gaussian Process optimizer (GP optimizer) [4]; adaptive Tree-structured Parzen Estimator (TPE) [4], instead of modeling the  $p(y|\lambda)$ , it applies a tree-structure Parzen estimator to concurrently model  $p(y|\lambda \geq a)$  and  $p(y|\lambda < a)$ , where  $a$  is a moving threshold; lastly, Bayesian Optimization and HyperBand (BOHB) [11] resembles multiple TPE models as its back-end surrogate. Other eye-catching examples are: Zaefferer et al. [38] demonstrated the feasibility of using Radial Basis Function networks (RBFn) to solve combinatorial optimization problems; Support Vector Regression (SVR) and gradient boosting regression tree are applied in [10] to model machine learning tasks. To come to the point, all the mentioned surrogate models can naturally handle or be modified to handle discrete variables (e.g. binary, ordinal, and categorical) or structures (tree, graph, etc).

## 3 SAMA-DiEGO

A novel algorithm, namely, the Self-Adaptive Multi-surrogate Assisted Discrete Efficient Optimization Algorithm (SAMA-DiEGO) is discussed in this section. Firstly, the formulation of the optimization task is given. The algorithm is proposed to solve a single objective optimization problem. For minimization problems<sup>2</sup>, the optimization task  $f: \mathcal{S} \rightarrow \mathbb{R}$ , can be defined as follows:

$$\mathbf{X} = \{x \mid \arg \min_{x \in \mathcal{S}} f(x), f(x) \in \mathbb{R}\}, \quad (1)$$

where  $\mathcal{S}$  is the search space that contains all feasible solutions and  $\mathbf{X}$  are the optimal solution(s). On that basis, a competent data-driven surrogate model  $\hat{f}$  to  $f: \mathcal{S} \rightarrow \mathbb{R}$  is expected to bear the following properties,

$$\begin{aligned} \hat{F} &= \{f(x) \mid x \in \hat{\mathcal{S}}, \hat{\mathcal{S}} \subseteq \mathcal{S}\}, \\ \hat{f}_{\hat{\mathcal{S}}, \hat{F}}(x) &: \mathcal{S} \rightarrow \mathbb{R}, \\ \exists \mathcal{S}^* \subseteq \mathcal{S}: |\mathcal{S}^*| > 1, \forall \hat{x} \in \mathcal{S}^*: \hat{f}_{\hat{\mathcal{S}}, \hat{F}}(\hat{x}) &\simeq f(x), \end{aligned} \quad (2)$$

where  $\hat{\mathcal{S}}$  is the set of available observable input from the solution space  $\mathcal{S}$  and their corresponding real objective values of  $\hat{\mathcal{S}}$  are recorded in  $\hat{F}$ . The surrogate model  $\hat{f}_{\hat{\mathcal{S}}, \hat{F}}$  learns (e.g. curve fitting) the data ( $\hat{\mathcal{S}}$  and  $\hat{F}$ ) and therefore, produces simulated objective values of  $f$  on some solutions ( $\mathcal{S}^*$ ) from  $\mathcal{S}$ . If the surrogate model can accurately simulate the original functions, i.e., given the same inputs, the surrogate model can generate similarly or even the same outputs as the original functions using less computational resources, it is possible to efficiently locate the global optimum of the original functions through exclusively doing optimization using the surrogate as a computationally cheap substitute objective.

Following the notations of the problem formulation, a pseudocode of the SAMA-DiEGO algorithm is presented in Alg. 1. The algorithm consists of two stages in general, namely an initialization stage and an optimization stage. The initialization stage determines the initial data sample and based on the initial samples to select the pool of candidate surrogate models for the next stage. The optimization stage iteratively decides the best surrogate model, searches for a promising solution on the determined model, and evaluates the solution using the real objective function. Before diving into details, explanations of three important choices made for SAMA-DiEGO are first discussed. These choices distinguish SAMA-DiEGO from the prototype [23], and other EGO algorithms (e.g. SMAC [21]).

**Infill Criterion:** Generally speaking, most of the Bayesian optimization rely on a probabilistic-based infill criterion (a.k.a. acquisition function) to find promising points [36]. One of the well-known and commonly used infill criteria is the Expected Improvement (EI). Assume given a surrogate model  $g(x)$  computed on a minimization problem and a

<sup>2</sup>A maximization problem can be transformed into a minimization problem by multiplying the objective value with -1.

reasonable assumption that the model follows a normal distribution, the definition of EI can be written as follows:

$$EI_g(\mathbf{x}) = (g^* - g(x))\Phi\left(\frac{g^* - g(x)}{\sigma(g(x))}\right) + \sigma(g(x))\phi\left(\frac{g^* - g(x)}{\sigma(g(x))}\right), \quad (3)$$

where  $g^*$  is the best-so-far found objective value,  $g(x)$  and  $\sigma(g(x))$  are the prediction and uncertainty of surrogate  $g$  at an input point  $x$ , e.g., the mean and standard deviation of a Gaussian process.  $\Phi$  and  $\phi$  are the cumulative distribution function and probability distribution function of a standard normal random variable. The motivation behind consulting a probabilistic-based infill criterion in EGO is to balance the exploration and exploitation of an optimization process such that the algorithm is unlikely to be trapped by local optima. However, with the assistance of multiple surrogate models, each portraying and capturing distinct landscapes of the objective function, a feasible alternative infill criterion, in comparison with the probabilistic-based ones used in single surrogate-assisted EGO, is the prediction value (PV) of the most promising surrogate model. It can be inferred that the prediction value of a surrogate becomes informative if the promising model being optimized changes drastically iteration-wise since the back-end optimizer can constantly explore different landscapes of the objective function. Rehbach et al. [29] experimentally discovered that exclusively using prediction values of a predetermined surrogate can outperform using expected improvement in handling higher dimensional (e.g., 10-dimensional) continuous problems. A similar experiment is conducted that compares the performance of SAMA-DiEGO with two infill criteria, i.e., prediction value (PV) and expected improvement (EI), obtained on numerous discrete problems to further discuss the necessity of using a probabilistic-based infill criterion in multiple surrogate-assisted discrete optimizations.

**Back-end Optimizer:** In Bayesian optimization, it is expected that the back-end optimizers can efficiently locate the optima of the infill criterion. When handling discrete problems, the non-differentiable search spaces preclude the usage of well-established gradient-based optimizers (e.g., the BFGS algorithm) but favor the usage of robust discrete optimizers (e.g., evolutionary algorithms). Notable non-gradient based optimizers are random local search in [21], Genetic Algorithm in [38] and Mixed-Integer Evolutionary Strategy (MIES) [26] in [32] and [19]. Following their successes, the SAMA-DiEGO employs MIES for its strong capability of handling both categorical and ordinal variables, which are the two most commonly seen types of discrete variables [13].

**Model Selection Mechanism:** Multiple online model management strategies are discussed in Section 2.1. Our intention of using multiple surrogates is to allow the optimizer to explore different approximations of the search space provided by distinct (types of) models without requiring additional calls to the objective function. It can be anticipated that auto-tuning strategies (e.g., [15]) become cumbersome since they have to cope with a considerable number of hyperparameters at a time. Moreover, the lack of existing studies on model-agnostic uncertainty quantification for discrete surrogate models also precludes the use of ensembles of models (e.g., [17]). Additionally, strategies to simultaneously use all surrogate models to find and evaluate promising solutions (e.g., MSEGO [35]) tend to waste expensive function evaluations if some surrogates constantly perform poorly. Therefore, an online model selection technique as proposed in [1] is incorporated in SAMA-DiEGO. Note that in our case, the sliding window mechanism is left out. The sliding window smooths the performance of surrogates across a few iterations and therefore leads to consistency in determining the best surrogate. However, this consistency in model selection contradicts the purpose of applying multiple surrogates in SAMA-DiEGO, and hence the mechanism is excluded.

### 3.1 A Walk-through of SAMA-DiEGO

After reasoning about the choices made for SAMA-DiEGO. A step-by-step introduction to SAMA-DiEGO is provided, starting by explaining the initialization stage, which consists of the following two steps:

- 1) Determine the initial design and evaluate these designs on the objective function (lines 2 and 3 in Alg. 1). In the absence of historical data, heuristic sampling is performed to obtain the initial samples. Specifically, Latin Hypercube Sampling (LHS) was chosen as suggested for discrete problems in [19] (see section 4.3)
- 2) With the initial samples, the algorithm verifies all available candidate surrogate models to rule out relatively poorly performing ones according to Feasibility and Performance (line 4 in Alg. 1):

**Feasibility:** Stork et al. [33] identified a new challenge as the feasibility of existing models. Therefore, with the initial data samples the surrogates are validated and the ineffectual ones are dropped. (e.g., indefinite Kriging kernels).

**Performance:** The applied ranking criterion  $C$  is a train-test split  $R^2$  validation. Models are fitted with training samples (70% of total) and ranked regarding their obtained  $R^2$  scores on the validation samples (the 'remaining' 30%). Meanwhile, all candidate models that require more than  $T$  seconds to converge are disqualified. As a result, the top  $P$  surrogates are chosen.

**Algorithm 1:** SAMA-DiEGO.

**Input:** Objective functions  $f(\mathbf{x})$ , decision parameter space  $\mathbf{S}$ , number of initial samples  $K$ , maximum evaluation budget  $N_{max}$ , model configurations  $M(\varphi)$ , number of available parallelisms  $P$ , infill criterion  $I$ , time limit  $T$ .

**Output:** Best found solution  $\mathbf{x}_{best}$  and its corresponding objective value  $y_{best}$ .

---

```

1 Function Main( $f, \mathbf{S}, K, N_{max}, M, I, P, T$ ):
2    $\mathbf{X} \leftarrow \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$  ▷ Generate initial design,  $\mathbf{x}_i \in \mathbf{S}$ 
3    $\mathbf{Y} \leftarrow f(\mathbf{X})$  ▷ Obtain objective scores,  $\mathbf{Y} \in \mathbb{R}^K$ 
4    $\tilde{F}\tilde{F}, M^* \leftarrow \text{VERIFICATION}(M, \mathbf{X}, \mathbf{Y}, P, T)$  ▷ Verification of surrogates
5    $\hat{m} \leftarrow \text{GETFIRSTOF}(M^*)$  ▷ Get the configurations of the best model after verification
6    $n \leftarrow K$ 
7   while  $n \leq N_{max}$  do
8      $\tilde{F}\tilde{F} \leftarrow []$  ▷ Initialize a pool of surrogates
9      $\hat{\mathbf{Y}} \leftarrow \text{STANDARDIZE}(\mathbf{Y})$ 
10    for  $\varphi \in M^*$  do ▷ For each model configuration
11       $\hat{f}_{\mathbf{X}, \hat{\mathbf{Y}}} \leftarrow \text{FIT}(\varphi, \mathbf{X}, \hat{\mathbf{Y}})$  ▷ Create and fit models
12       $\tilde{F}\tilde{F} \leftarrow [\tilde{F}\tilde{F} \hat{f}_{\mathbf{X}, \hat{\mathbf{Y}}}]$  ▷ Add models into pool
13    end
14     $\bar{y} \leftarrow \text{MEAN}(\mathbf{Y})$  ▷ Get the mean of Objective Values
15     $\sigma \leftarrow \text{STD}(\mathbf{Y})$  ▷ Get the standard deviation
16     $\mathbf{y}_{best} \leftarrow \text{UPDATEBEST}(\mathbf{Y})$  ▷ Update the best-so-far
17     $\mathbf{y}_{best}^* \leftarrow (\mathbf{y}_{best} - \bar{y}) / \sigma$  ▷ Standardization
18     $\hat{f}_{\mathbf{X}, \hat{\mathbf{Y}}}^* \leftarrow \text{GETMODELBYCONFIG}(\tilde{F}\tilde{F}, \hat{m})$  ▷ Get the best model by configuration
19     $\mathbf{IC} \leftarrow I(\hat{f}_{\mathbf{X}, \hat{\mathbf{Y}}}^*, \mathbf{y}_{best}^*)$  ▷ Configure the infill criterion
20     $\mathbf{x}_n \leftarrow \text{OPTIMIZE}(\mathbf{IC}, \hat{f}_{\mathbf{X}, \hat{\mathbf{Y}}}^*, P, \mathbf{X})$  ▷ Find best new solution w.r.t  $\mathbf{IC}$ 
21     $\mathbf{y}_n \leftarrow f(\mathbf{x}_n)$  ▷ Evaluate the new solution
22     $\mathbf{y}_n^* \leftarrow (\mathbf{y}_n - \bar{y}) / \sigma$  ▷ Standardize the new sample
23     $\hat{f}_n \leftarrow \text{RANKMODEL}(\tilde{F}\tilde{F}, \mathbf{x}_n, \mathbf{y}_n^*)$  ▷ Rank and select the best model
24     $\hat{m} \leftarrow \text{GETCONFIGOFMODEL}(\hat{f}_n)$ 
25     $\mathbf{X} \leftarrow [\mathbf{X} \mathbf{x}_n]$  ▷ Add the new solution,  $\mathbf{X} \in \mathbb{R}^{d \times (n+1)}$ 
26     $\mathbf{Y} \leftarrow [\mathbf{Y} \mathbf{y}_n]$  ▷ Add the objective value,  $\mathbf{Y} \in \mathbb{R}^{n+1}$ 
27     $n \leftarrow n + 1$ 
28  end
29   $\mathbf{x}_{best}, \mathbf{y}_{best} \leftarrow \text{FINDBEST}(\mathbf{X}, \mathbf{Y})$  ▷ The best w.r.t.  $\mathbf{Y}$ 
30 return ( $\mathbf{x}_{best}, \mathbf{y}_{best}$ )

```

---

The verification phase yields a pool of promising surrogate models maintained according to their rankings in the first stage, i.e., the best-performing model and its configuration (parameters) is the first of the  $\tilde{F}\tilde{F}$  and  $M^*$ . These models are subsequently utilized in the second stage (the optimization), which is enclosed in a while-loop in lines 7-27 of Alg. 1. Being fundamentally based on the Efficient Global Optimization algorithm, the optimization loop consists of five steps:

1) Standardize all obtained objective values  $\mathbf{Y}$  as follows:

$$\hat{\mathbf{Y}} = \frac{\mathbf{Y} - \bar{y}}{\sigma}, \quad (4)$$

where  $\bar{y}$  and  $\sigma$  are the mean and standard deviation of  $\mathbf{Y}$  respectively, and  $\hat{\mathbf{Y}}$  is the output (line 9).

2) Create and fit candidate surrogate models with standardized data samples (i.e.,  $\mathbf{X}$  and  $\hat{\mathbf{Y}}$ ) (lines 10-13). This process is done in parallel, where  $P$  surrogate models are concurrently fitted with the same data in  $P$  processes.

3) Set up the acquisition function based on the previously selected surrogate model and the standardized best-so-far objective score (lines 14-19). The determined surrogate model is the best-performing one in the previous iteration (see step 5) or the best in the verification stage in the case of the first iteration of the optimization loop.

4) Search and evaluate the next evaluation point (line 20). Through optimizing the selected infill criterion from step 3, i.e., maximizing EI or finding the optimal PV, the SAMA-DiEGO locates the next promising solution (line 20) and obtains its real objective value regarding the objective function (line 21).

5) Rank surrogate models and select the best. The procedure for determining the best surrogate model  $\hat{f}_n$  based on the  $n$ th evaluated data sample in the  $n$ th iteration is as follows:

$$\hat{f}_n = \arg \min_{\hat{f}_{\mathbf{X}, \mathbf{Y}} \in \tilde{F}F} |\hat{f}_{\mathbf{X}, \mathbf{Y}}(\mathbf{x}_n) - \mathbf{y}_n^*|, \quad (5)$$

where  $\tilde{F}F$  is the pool of trained surrogate models,  $\mathbf{x}_n$  and  $\mathbf{y}_n^*$  are the  $n$ th evaluated solution and its real objective value respectively.

The above optimization loop is repeated until the maximum allowed number of evaluations on the objective function (budget) is reached. Lastly, the algorithm returns the best-found evaluated solution  $X_{best}$  and its real objective value,  $y_{best}$ .

## 4 Experimental Setup

Experiments have been carried out to validate the newly proposed SAMA-DiEGO algorithms on discrete benchmark problems, and before diving into the results, setup of these experiments is first discussed in this section.

### 4.1 The Used Surrogate Modeling Techniques

A total of thirty-one distinct surrogate models that naturally support or can be realistically extended to support discrete variables are used in this study. These models belong to four surrogate classes:

**Radial Basis Function network (RBFn):** The augmented RBFn applied in [8] are chosen for its unrestricted uncertainty quantification which enables us to compute EI on an arbitrary RBFn. Let  $\varphi(d)$  be a radial basis function defined on a Euclidean-distance-based measurement  $d$ . Nine radial basis functions, including *linear* with  $\varphi(d) = d$ , *polyharmonic 4* with  $\varphi(d) = d^4 \cdot \log(d)$ , *polyharmonic 5* with  $\varphi(d) = d^5$  and other six functions that have been used in [8], were experimented in this research.

**Kriging interpolation (Kriging):** A universal Kriging model (a.k.a. Gaussian process regression) is a linear combination of a basis function added to a random process with zero mean and a covariance function [7]. A convenient implementation of Kriging models provided in the SMT toolbox [7] is applied in our experiments. This implementation realized a novel transformation mechanism that was proposed in [13], which practically allows Kriging to model discrete problems. Consequently, five correlation functions are selected, namely *Ornstein–Uhlenbeck process*, *Squared Gaussian correlation*, *Matérn 3/2*, *Matérn 5/2*, and *Gower autocorrelation*, from the SMT toolbox. Each covariance function is then combined with one of the three basis functions (*constant*, *linear*, and *quadratic*) to form a Kriging model, and hence 15 in total.

**Random Forest regression (RF):** Random forest is an ensemble of decision trees, and it uses the mean of the predictions of all trees to be its prediction. Our utilized RF followed the implementation of van Stein et al. [32], and it can quantify the uncertainty of a prediction by calculating the variance of the predictions of ensembled decision trees. Hence, this permits us to compute the expected improvement in RF.

**Support-Vector Machines regression (SVM):** SVM aims to determine a set of hyperplanes that maximizes the linear separability of data. Given data that is non-linearly separable, the SVM uses kernel functions (e.g., RBF and polynomial) to map data to higher dimensionality. The well-established  $\epsilon$ -SVM implementation is chose of scikit-learn [27] (v0.24.2) in this research. Six kernels, namely *linear*, *rbf* and *sigmoid*, and three *polynomial* kernels with 2, 3, and 5 degrees, are considered in this research. To the best of our knowledge, studies on quantifying the uncertainty for SVMs do not exist, hence the backend optimizer searches for the next evaluation point exclusively regarding the predictions of SVM regression.

To summarize, the surrogate pool of SAMA-DiEGO contains nine RBFn, fifteen Kriging, one RF regression, and six SVM regression models. The hyperparameters of all these thirty-one models are not specifically tuned but set according to settings in original implementations or literature.

## 4.2 Benchmarked Problems

Discrete problems can be divided into two classes in regards to the type of variables: ordinal (e.g., integer-valued) and combinatorial (e.g., binary) [13]. With regard to the two classes, two experiments are designed, each representing a type of variable, to benchmark the performance of SAMA-DiEGO against several robust optimization algorithms.

1) The first problem set is sampled from the Black Box Optimization Benchmark (BBOB) *minimization* problems [18], where the discretization method proposed in [34] was applied to transform the continuous input space into an ordinal one. To be distinguished from combinatorial problems with large dimensionality, the search space is configured to be limited in dimensionality but abundant in choices per variable. Moreover, a preliminary study was carried out to categorize the problems into three groups regarding the fidelity of surrogate models [20]. Consequently, fifteen problems are randomly and evenly selected from the three groups and considered in dimensionality 15 with 101 ordinal choices (0 to 100) per variable:

- 1 : F6 Attractive Sector Function
- 2 : F7 Step Ellipsoidal Function
- 3 : F9 Rosenbrock Function, rotated
- 4 : F8 Rosenbrock Function
- 5 : F11 Discus Function
- 6 : F13 Sharp Ridge Function
- 7 : F15 Rastrigin Function
- 8 : F16 Weierstrass Function
- 9 : F18 Schaffers F7 Function, moderately ill-conditioned
- 10 : F19 Composite Griewank-Rosenbrock Function F8F2
- 11 : F20 Schwefel Function
- 12 : F23 Katsuura Function
- 13 : F21 Gallagher’s Gaussian 101-me Peaks Function
- 14 : F22 Gallagher’s Gaussian 21-hi Peaks Function
- 15 : F24 Lunacek bi-Rastrigin Function

2) The second set is selected from the IOH Pseudo-Boolean Optimization (PBO) benchmark [9], which features 23 well-organized binary-encoded combinatorial *maximization* problems. Five problems were chosen from the benchmark following the recommendations of [19]:

- LABS** : low auto-correlation binary sequences problem
- Ising1D Ring** Ising model defined on one-dimensional ring
- Ising2D Torus** Ising model defined on a two-dimensional torus
- MIVS** : maximum independent vertex set problem
- NQP** : N-queens problem

The dimensionalities of problems are determined also in compliance with the settings of [19], which are {25, 64, 100}. The global optima of the five functions are known regardless of dimensionality.

## 4.3 Benchmarked Algorithms

The benchmarked algorithms are listed in Table 1. The first algorithm, MIES, is the backend optimizer of both MIP-EGO and the proposed SAMA-DiEGO. It serves as the very baseline algorithm for the ordinal BBOB problems. MIP-EGO [32], adaptive TPE [5] and SMAC [21] are three efficient and widely-used single-surrogate-assisted EGO algorithms that can solve both discrete and continuous problems. P3-GOMEA is a state-of-the-art version of the gene-pool optimal mixing algorithm that can tackle discrete problems [10]. Four types of algorithms were selected specifically for IOH pseudo-boolean problems, where the  $(1 + \lambda)$  EA and  $(1+(\lambda, \lambda))$  GA are existing robust non-surrogate-assisted algorithms suggested by [9]. Specifically, each of the five experimented EAs features a distinct strategy to control mutation rates. CatES and SVM-CatES were experimented by Horesh et al. [19] on the same five IOH pseudo-boolean problems (with the same settings), therefore our results can directly be compared to the results of [19]. Lastly, two proposed SAMA-DiEGO algorithms were experimented and compared to analyze the necessity of using probabilistic-based infill criterion in this context. The (hyper)parameters of all algorithms, except for SAMA-DiEGO, were set in compliance with their publically available implementations and referential literature. Moreover,

Table 1: The benchmarked algorithms.

Algorithm	Surrogate	Specification			Experiment	
MIES	None	Mixed-Integer Evolutionary Strategy [26].			First	
Adaptive TPE	TPE	Adaptive Tree-structured Parzen Estimator approach [5]			First	
SMAC	RF	Sequential Model Algorithm Configuration [21]			First	
MIP-EGO	RF	Mixed-Integer Parallel Efficient Global Optimization [32]			Both	
P3-GOMEA	None	Parameterless Population Pyramid Gene-Pool Optimal Mixing Algorithm [10]			First	
(1 + 10) EA	None	Five (1 + $\lambda$ ) evolutionary algorithms with a population size of 10. [9]			Second	
(1+(25, 25)) GA	None	A self-adjusting genetic algorithm with a population size of 25. [9]			Second	
CatES	None	Categorical evolutionary strategy [19]			Second	
SVM-CatES	SVM	Support Vector Machines regression assisted CatES [19]			Second	
SAMA-DiEGO	9 RBF, 6 SVMs, 15 Kriging, 1 RF	Type	Hyperchoices		-	
			Infill Criterion ( $I$ )	$P$		$T$
		PV	Prediction Value	7 parallelisms		30 seconds
	EI	Expected Improvement			Both	

Surrogate: the type(s) of surrogate model used by the algorithm;

Experiment: on which problem set(s) do(es) the algorithm(s) got benchmarked, *first* for IOH problems, *second* for discretized BBOB, and *both* for both problem sets.

the two hyperparameters of SAMA-DiEGO (i.e.,  $P$  and  $T$  in Alg. 1) were exclusively determined with respect to our available computational resources (i.e., octa-processes per usage).

Apart from individual hyperparameters of each algorithm, a parameter not to be neglected is the size of initial samples for all surrogate-assisted algorithms. For the IOH pseudo-boolean problems, the initial sampling size was set to the dimensionality of the problem, which was primarily bounded by the applied Latin hypercube sampling strategy of SAMA-DiEGO. And following the settings of [29], the size of initial sampling is set to 10% of the total budget (function calls to the objective function) for the experiments on discretized BBOB problems.

With regard to the aforementioned setups, a fixed-budget experiment is proposed to analyze and compare the performance of the algorithms. All algorithms are allowed to use up to 500 function calls for a given optimization problem. Moreover, each algorithm was independently executed 11 times on a given problem to (maximally) stabilize the performance under our computational resources.

## 5 Results

### 5.1 The Ordinal BBOB Problems

The benchmark results of algorithms on the discretized BBOB *minimization* problems are shown in Table 2, Table 3, and Table 4. Additionally, an analysis on the convergence profiles of this experiment can be found in [20].

The results in Table 2 are the cases where SAMA-DiEGO with prediction value is generally the best algorithm. Chasing after SAMA-DiEGO-PV, SMAC becomes a competitive runner-up on two test problems (i.e., F6, F11). Next to Table 2, first Table 3 is presented where multiple algorithms give comparable results. SAMA-DiEGO with expected improvement becomes twice the best algorithms regarding the best mean objective value, and adaptive TPE leads on F18. Lastly, Table 4 shows cases where SMAC tops the results, and SAMA-DiEGO arrives second on four problems.

Through relating the performance of algorithms to properties of problems as mentioned in [18], it can be observed that our proposed SAMA-DiEGO is a robust algorithm for single-modal problems but shows a slight deficiency for multi-modal problems in comparison with SMAC. These observations are reasonable since the rapid switch of surrogate models in SAMA-DiEGO helps the algorithm converge on problems with clear global structures, but inevitably distracts the search when handling problems with a considerable number of local structures spread over the whole search space (e.g., F21 and F22). Furthermore, it can also be discovered that prediction value is generally the better infill criterion for SAMA-DiEGO, but expected improvement should be favored if the problems are multi-modal across the entire search space. Lastly, by comparing the results of MIES (non-surrogate), MIPEGO (single-surrogate), and SAMA-DiEGO (multi-surrogate), it can be concluded that using a surrogate model indeed facilitates the optimization of ordinal-spaced problems. And moreover, our proposed strategy, which considers multi-surrogate models, can benefit the search process even more.

### 5.2 The Pseudo-Boolean Problems

The results of the second experiment are presented in Table 5. The best-performing algorithms are highlighted in blue if they outperform others in a Wilcoxon rank sum test ( $\alpha = 0.05$ ) over 11 independent runs. According to



Table 2: Benchmark results on discretized (ordinal) BBOB minimization problems led by **SAMA-DiEGO**. For each test problem, **blue** highlights the algorithm that achieved the lowest average best-found fitness value across 11 independent runs. Whereas, the other results highlighted in **green** are incomparable to the best solution according to a Wilcoxon rank sum test with  $\alpha = 0.05$ .

Function	Algorithm	Dimension = 15			
		Fitness Value			
		Best	Mean	Median	Std
F6 Attractive Sector	MI-ES	172869.15	407604.29	428103.05	209158.25
	MIP-EGO	21227.12	61558.02	53880.90	13787.24
	Adaptive TPE	387.80	6884.71	7191.29	5945.59
	<b>SMAC</b>	<b>72.05</b>	<b>99.15</b>	<b>99.16</b>	<b>16.20</b>
	P3-GOMEA	148.02	3884.69	12891.50	18872.99
	<b>SAMA-DiEGO-PV</b>	<b>76.35</b>	<b>93.44</b>	<b>94.86</b>	<b>17.81</b>
	SAMA-DiEGO-EI	233.04	11822.81	14705.90	10733.29
F7 Step Ellipsoidal	MI-ES	361.10	682.51	765.20	314.34
	MIP-EGO	223.94	269.91	276.73	32.11
	Adaptive TPE	139.31	164.78	177.11	25.68
	<b>SMAC</b>	<b>111.35</b>	<b>126.16</b>	<b>128.40</b>	<b>11.47</b>
	P3-GOMEA	156.84	209.53	206.48	28.50
	<b>SAMA-DiEGO-PV</b>	<b>104.07</b>	<b>114.74</b>	<b>117.48</b>	<b>11.39</b>
	SAMA-DiEGO-EI	146.07	180.11	188.11	31.78
F8 Rosenbrock Original	MI-ES	43955.62	104023.45	111845.50	46027.14
	MIP-EGO	14296.34	17477.50	17779.66	2520.72
	Adaptive TPE	1302.25	4796.49	5614.59	2990.26
	<b>SMAC</b>	<b>565.94</b>	<b>2225.38</b>	<b>4038.31</b>	<b>4345.35</b>
	P3-GOMEA	42008.57	59620.57	69886.26	25249.70
	<b>SAMA-DiEGO-PV</b>	<b>283.74</b>	<b>606.00</b>	<b>609.30</b>	<b>191.79</b>
	SAMA-DiEGO-EI	1073.39	5357.91	4928.82	2720.72
F9 Rosenbrock Rotated	MI-ES	44756.84	73232.62	79370.41	31158.36
	MIP-EGO	5123.26	14679.66	13981.47	5139.36
	Adaptive TPE	1401.03	2181.28	2284.56	753.30
	<b>SMAC</b>	<b>500.69</b>	<b>2478.33</b>	<b>2293.00</b>	<b>1502.34</b>
	P3-GOMEA	579.97	1547.67	2241.56	1597.13
	<b>SAMA-DiEGO-PV</b>	<b>214.83</b>	<b>256.11</b>	<b>316.88</b>	<b>159.97</b>
	SAMA-DiEGO-EI	214.83	426.35	581.77	328.22
F11 Discus	MI-ES	206.25	279.45	207713.44	452104.85
	MIP-EGO	166.11	233.11	337.58	276.67
	Adaptive TPE	125.13	217.01	229.29	78.24
	<b>SMAC</b>	<b>135.17</b>	<b>187.86</b>	<b>199.90</b>	<b>41.14</b>
	P3-GOMEA	160.71	205.91	207.61	27.70
	<b>SAMA-DiEGO-PV</b>	<b>152.51</b>	<b>179.92</b>	<b>185.90</b>	<b>19.26</b>
	SAMA-DiEGO-EI	168.88	215.24	217.75	30.85
F19 Composite Griewank-Rosenbrock	MI-ES	-82.41	-70.90	-70.03	11.00
	MIP-EGO	-92.40	-90.98	-90.82	1.12
	Adaptive TPE	-96.42	-95.68	-95.44	0.79
	<b>SMAC</b>	<b>-97.05</b>	<b>-94.95</b>	<b>-94.90</b>	<b>1.03</b>
	P3-GOMEA	-96.13	-94.03	-94.05	1.20
	<b>SAMA-DiEGO-PV</b>	<b>-102.30</b>	<b>-96.38</b>	<b>-96.91</b>	<b>2.11</b>
	SAMA-DiEGO-EI	-96.90	-94.76	-95.00	0.87
F20 Schwefel	MI-ES	25393.04	72963.97	76338.17	33981.88
	MIP-EGO	1580.41	5264.21	5734.52	2624.57
	Adaptive TPE	-514.11	-128.67	66.48	474.78
	<b>SMAC</b>	<b>-543.55</b>	<b>298.68</b>	<b>1000.94</b>	<b>1143.17</b>
	P3-GOMEA	-542.92	-201.39	564.72	1157.05
	<b>SAMA-DiEGO-PV</b>	<b>-544.35</b>	<b>-543.55</b>	<b>-543.59</b>	<b>0.40</b>
	SAMA-DiEGO-EI	-535.31	-366.33	-193.45	416.77

Table 3: Benchmark results on discretized (ordinal) BBOB minimization problems. For each test problem, blue highlights the algorithm that achieved the lowest average best-found fitness value across 11 independent runs. Whereas, the other results highlighted in green are incomparable to the best solution according to a Wilcoxon rank sum test with  $\alpha = 0.05$ .

Function	Algorithm	Dimension = 15			
		Best	Mean	Median	Std
F18 Schaffers F7 (moderately ill-conditioned)	MI-ES	15.54	29.97	31.97	11.10
	MIP-EGO	0.80	17.16	17.39	7.09
	Adaptive TPE	-9.57	0.11	-0.56	3.12
	SMAC	-9.13	0.31	0.48	6.99
	P3-GOMEA	0.65	6.62	5.86	3.06
	SAMA-DiEGO-PV	-3.17	1.19	1.04	2.93
	SAMA-DiEGO-EI	-4.82	0.44	1.02	4.57
F23 Katsuura	MI-ES	9.92	11.13	11.21	0.69
	MIP-EGO	8.99	10.65	10.34	0.75
	Adaptive TPE	9.50	10.63	10.52	0.76
	SMAC	9.20	10.55	10.65	0.58
	P3-GOMEA	9.03	10.30	10.27	0.53
	SAMA-DiEGO-PV	9.85	10.81	10.86	0.68
	SAMA-DiEGO-EI	9.64	10.16	10.30	0.60
F24 Lunacek bi-Rastrigin	MI-ES	358.86	437.38	456.27	71.83
	MIP-EGO	282.88	320.01	319.88	15.99
	Adaptive TPE	253.99	280.45	276.14	10.94
	SMAC	246.68	279.30	288.88	33.66
	P3-GOMEA	276.85	295.09	294.50	11.66
	SAMA-DiEGO-PV	245.91	303.27	297.76	31.59
	SAMA-DiEGO-EI	229.96	276.68	276.03	21.78

Table 5, SAMA-DiEGO performs best on fourteen of the fifteen test cases, with the exception of the 64-dimensional MIVS problems, where the (1+10) EA performed better. Moreover, as it is indicated in **Budgets Mean** columns of the result table, both SAMA-DiEGOs can find the global optimum of the two Ising problems across three dimensionalities and are also capable of solving NQP and MIVS on 25 dimensions within 500 fitness evaluations. Unexpectedly, the single surrogate assisted algorithms MIP-EGO performed worse than the (1+10) evolutionary algorithms in fourteen of the fifteen cases. Furthermore, because SAMA-DiEGO, SVM-CatES, and MIP-EGO all use MI-ES as a back-end solver, the differences in performance of the three algorithms can be attributed to their respective surrogate models, substantiating the benefit of using our proposed model management strategy. Also, SAMA-DiEGO with EI did better on 64 and 100 dimensions than SAMA-DiEGO with PV. Therefore, it is reasonable to consider that the expected improvement is a booster for SAMA-DiEGO to solve higher dimensional combinatorial problems in comparison with its sibling.

## 6 Conclusions and Future works

In this paper, a novel algorithm, Self-Adaptive Multiple-surrogate Assisted Efficient Global Optimization (SAMA-DiEGO), is proposed. Following the iterative optimization process of Bayesian optimization, the SAMA-DiEGO additionally features a two-stage online model selection mechanism, which iteratively chooses the most suitable surrogate from a pool of candidate models to serve as the cheaper substitute for the discrete objective function. Experiments were carried out to compare SAMA-DiEGO with existing robust optimization algorithms on fifteen binary-encoded combinatorial test cases and fifteen ordinal-encoded problems. The results indicated that SAMA-DiEGO is a robust solver for combinatorial problems, and ordinal problems. Moreover, the performance of SAMA-DiEGO with expected improvement as an infill criterion was compared with that of exclusively using the prediction value of the promising surrogate. In conclusion, prediction value is an efficient and promising infill criterion for multi-surrogate assisted EGO

Table 4: Benchmark results on discretized (ordinal) BBOB problems led by **SMAC**. For each test problem, **blue** highlights the algorithm that achieved the lowest average best-found fitness value across 11 independent runs. Whereas, the other results highlighted in **green** are incomparable to the best solution according to a Wilcoxon rank sum test with  $\alpha = 0.05$ .

Function	Algorithm	Dimension = 15			
		Fitness Value			
		Best	Mean	Median	Std
F13 Sharp Ridge	MI-ES	1805.62	2177.03	2186.14	225.16
	MIP-EGO	943.29	1009.77	1042.65	87.45
	Adaptive TPE	668.11	758.38	762.45	84.39
	<b>SMAC</b>	<b>258.09</b>	<b>417.76</b>	<b>412.22</b>	<b>88.60</b>
	P3-GOMEA	656.29	856.28	855.72	145.99
	<b>SAMA-DiEGO-PV</b>	<b>268.56</b>	<b>514.22</b>	<b>492.19</b>	<b>122.41</b>
	<b>SAMA-DiEGO-EI</b>	<b>625.11</b>	<b>827.61</b>	<b>863.90</b>	<b>146.71</b>
F15 Rastrigin Moderately Ill-conditioned	MI-ES	1337.55	1544.87	1572.05	223.00
	MIP-EGO	1231.29	1341.62	1327.01	51.03
	Adaptive TPE	1173.00	1198.76	1201.05	20.33
	<b>SMAC</b>	<b>1098.96</b>	<b>1170.15</b>	<b>1163.20</b>	<b>30.73</b>
	P3-GOMEA	1177.40	1217.94	1226.36	36.73
	<b>SAMA-DiEGO-PV</b>	<b>1123.21</b>	<b>1172.55</b>	<b>1167.24</b>	<b>23.65</b>
	<b>SAMA-DiEGO-EI</b>	<b>1172.35</b>	<b>1207.25</b>	<b>1217.00</b>	<b>34.37</b>
F16 Weierstrass	MI-ES	84.24	100.79	96.79	7.81
	MIP-EGO	89.66	96.05	96.33	3.50
	Adaptive TPE	87.63	98.08	96.99	3.23
	<b>SMAC</b>	<b>85.14</b>	<b>92.29</b>	<b>91.49</b>	<b>3.10</b>
	P3-GOMEA	91.99	96.09	96.96	3.46
	<b>SAMA-DiEGO-PV</b>	<b>85.10</b>	<b>93.22</b>	<b>93.25</b>	<b>5.70</b>
	<b>SAMA-DiEGO-EI</b>	<b>90.34</b>	<b>96.64</b>	<b>96.23</b>	<b>3.63</b>
F21 Gallagher's Gaussian 101-me Peaks	MI-ES	97.56	112.49	111.21	6.81
	MIP-EGO	75.47	91.13	91.16	9.26
	Adaptive TPE	53.32	60.82	62.80	8.61
	<b>SMAC</b>	<b>42.45</b>	<b>51.12</b>	<b>49.88</b>	<b>4.96</b>
	P3-GOMEA	51.45	72.57	73.36	15.09
	<b>SAMA-DiEGO-PV</b>	<b>49.80</b>	<b>71.38</b>	<b>74.09</b>	<b>12.93</b>
	<b>SAMA-DiEGO-EI</b>	<b>46.81</b>	<b>51.16</b>	<b>56.28</b>	<b>7.76</b>
F22 Gallagher's Gaussian 21-hi Peaks	MI-ES	-925.42	-917.58	-918.79	3.07
	MIP-EGO	-959.34	-937.17	-942.87	10.24
	Adaptive TPE	-977.57	-948.56	-952.38	12.52
	<b>SMAC</b>	<b>-997.29</b>	<b>-979.23</b>	<b>-982.75</b>	<b>10.64</b>
	P3-GOMEA	-980.78	-943.83	-952.91	14.21
	<b>SAMA-DiEGO-PV</b>	<b>-963.68</b>	<b>-939.61</b>	<b>-943.27</b>	<b>13.19</b>
	<b>SAMA-DiEGO-EI</b>	<b>-978.12</b>	<b>-971.07</b>	<b>-966.68</b>	<b>11.30</b>

Table 5: The experimental results obtained on five PBO problems defined on 25, 64, and 100 dimensions. The results are highlighted in blue if they are significantly better according to a Wilcoxon rank sum test with  $\alpha = 0.05$  over eleven independent runs.

Functions	Algorithms	Dimension = 25				Dimension = 64				Dimension = 100			
		Objective Value			Budgets Mean <sup>†</sup>	Objective Value			Budgets Mean <sup>†</sup>	Objective Value			Budgets Mean <sup>†</sup>
		Mean	Std	Best		Mean	Std	Best		Mean	Std	Best	
Ising1D	CatES	21.68	N/A	25.00	N/A	48.24	N/A	54.00	N/A	71.16	N/A	76.00	N/A
	SVM-CatES	23.40	N/A	25.00	N/A	53.56	N/A	56.00	N/A	77.44	N/A	84.00	N/A
	MIP-EGO	21.50	0.89	23.00	500	45.09	1.31	48.00	500	65.46	0.89	66.00	500
	(1+10) EA*	21.73	1.54	25.00	485	52.73	1.55	56.00	500	81.09	3.94	86.00	500
	(1+(25, 25)) GA	21.73	0.96	23.00	500	51.64	1.43	54.00	500	76.73	2.74	82.00	500
	SAMA-DiEGO-PV	25.00	0.00	25.00	259	64.00	0.00	64.00	99	100.00	0.00	100.00	125
SAMA-DiEGO-EI	25.00	0.00	25.00	291	64.00	0.00	64.00	66	100.00	0.00	100.00	106	
Ising2D	CatES	42.04	N/A	50.00	N/A	90.50	N/A	98.00	N/A	133.44	N/A	152.00	N/A
	SVM-CatES	49.92	N/A	50.00	N/A	101.64	N/A	116.00	N/A	144.44	N/A	154.00	N/A
	MIP-EGO	39.27	1.07	44.00	500	83.28	1.77	86.00	500	122.91	2.88	126.00	500
	(1+10) EA*	45.23	4.69	50.00	407	99.28	5.00	106.00	500	153.46	4.98	164.00	500
	(1+(25, 25)) GA	44.36	4.07	50.00	485	96.37	3.89	102.00	500	142.37	2.93	146.00	500
	SAMA-DiEGO-PV	50.00	0.00	50.00	150	128.00	0.00	128.00	85	200.00	0.00	200.00	123
SAMA-DiEGO-EI	50.00	0.00	50.00	177	128.00	0.00	128.00	68	200.00	0.00	200.00	110	
NQP	CatES	4.16	N/A	5.00	N/A	-36.96	N/A	-3.00	N/A	-264.30	N/A	24.00	N/A
	SVM-CatES	4.56	N/A	5.00	N/A	2.64	N/A	6.00	N/A	-15.20	N/A	26.00	N/A
	MIP-EGO	-0.73	2.67	3.00	500	-247.55	29.94	-193.00	500	-753.3	53.80	-641.00	500
	(1+10) EA*	4.36	0.64	5.00	392	3.55	5.25	8.00	500	-46.60	32.80	7.00	500
	(1+(25, 25)) GA	3.81	0.93	5.00	451	-59.55	22.92	-25.00	500	-288.60	73.70	-186.0	500
	SAMA-DiEGO-PV	5.00	0.00	5.00	195	3.36	0.48	4.00	500	6.00	1.00	7.00	500
SAMA-DiEGO-EI	5.00	0.00	5.00	260	6.00	0.43	7.00	500	6.30	0.60	7.00	500	
LABS	CatES	3.93	N/A	6.01	N/A	2.58	N/A	3.58	N/A	2.11	N/A	2.73	N/A
	SVM-CatES	4.31	N/A	6.51	N/A	2.76	N/A	4.03	N/A	2.24	N/A	2.94	N/A
	MIP-EGO	3.26	0.45	4.34	500	2.17	0.20	2.69	500	1.77	0.08	1.91	500
	(1+10) EA*	4.16	0.75	4.88	500	3.31	0.32	3.97	500	3.10	0.27	3.55	500
	(1+(25, 25)) GA	4.01	0.94	6.51	500	2.53	0.21	2.83	500	2.19	0.17	2.48	500
	SAMA-DiEGO-PV	4.80	0.78	6.51	500	2.71	0.21	3.03	500	2.77	0.31	3.32	500
SAMA-DiEGO-EI	4.10	0.62	5.58	500	3.69	0.21	4.23	500	3.17	0.34	3.74	500	
MIVS	CatES	10.84	N/A	12.00	N/A	-1.00	N/A	24.00	N/A	-308.60	N/A	-99.00	N/A
	SVM-CatES	11.66	N/A	12.00	N/A	17.44	N/A	23.00	N/A	-63.18	N/A	-3.00	N/A
	MIP-EGO	8.82	0.83	10.00	500	-370.36	127.11	-116.00	500	-1657.18	263.19	-1069.00	500
	(1+10) EA*	11.45	0.89	12.00	271	26.00	1.54	29.00	500	-21.73	77.64	36.00	500
	(1+(25, 25)) GA	10.63	1.30	12.00	431	-17.18	49.56	21.00	500	-353.36	251.72	23.00	500
	SAMA-DiEGO-PV	12.0	0.00	12.00	276	11.45	0.99	13.00	500	23.45	6.43	36.00	500
SAMA-DiEGO-EI	12.0	0.00	12.00	299	22.73	2.18	26.00	500	28.73	4.05	34.00	500	

<sup>†</sup> Each *Budgets Mean* column records the average number of function evaluations over 11 runs used by all algorithms to find the global optimum (if found, otherwise 500) of the problem respectively. Such results for ES and SVM-CatES are not available in case they are not provided in [19].

\* Instead of showing all results obtained by five (1 + 10) EA algorithms, the result of the best-performed (1 + 10) EA according to the significance test is shown per case.

algorithms in general, while expected improvement is better for solving high-dimensional combinatorial problems and highly multi-modal ordinal problems.

Lastly, several foreseeable directions following this research are briefly discussed. Firstly, it would be beneficial to track and compare the running time of algorithms in future benchmarking. Secondly, given the fact that all components of SAMA-DiEGO essentially work for both continuous and discrete variables, it would be a rewarding attempt to further investigate the benefit of using SAMA-DiEGO in mixed-integer cases. Other possible improvements are utilizing novel infill criterion (e.g. moment generating function of improvement [36]) to better tackle multi-modal problems, and employing hybrid dimensionality-reduction methods (e.g., PCA for BO [28]) to efficiently handle high-dimensional combinatorial problems.

## References

- [1] Samineh Bagheri, Wolfgang Konen, and Thomas Bäck. Online selection of surrogate models for constrained black-box optimization. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016.
- [2] Thomas Bartz-Beielstein and Martin Zaefferer. Model-based methods for continuous and discrete global optimization. *Applied Soft Computing*, 55:154–167, June 2017. Publisher: Elsevier.
- [3] Paul Beaucaire, Charlotte Beauthier, and Caroline Sainvitu. Multi-point infill sampling strategies exploiting multiple surrogate models. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1559–1567. Association for Computing Machinery, 2019.

- [4] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11*, pages 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc. event-place: Granada, Spain.
- [5] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, June 2013. PMLR. Issue: 1.
- [6] Atharv Bhosekar and Marianthi Ierapetritou. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, 108:250–267, January 2018. Publisher: Pergamon.
- [7] Mohamed Amine Bouhlel, John T. Hwang, Nathalie Bartoli, Rémi Lafage, Joseph Morlier, and Joaquim R. R. A. Martins. A Python surrogate modeling framework with derivatives. *Advances in Engineering Software*, 135:102662, 2019.
- [8] Roy de Winter, Bas van Stein, and Thomas Bäck. SAMO-COBRA: A fast surrogate assisted constrained multi-objective optimization algorithm. In Hisao Ishibuchi, Qingfu Zhang, Ran Cheng, Ke Li, Hui Li, Handing Wang, and Aimin Zhou, editors, *Evolutionary Multi-Criterion Optimization*, pages 270–282, Cham, 2021. Springer International Publishing.
- [9] Carola Doerr, Furong Ye, Naama Horesh, Hao Wang, Ofer M. Shir, and Thomas Bäck. Benchmarking discrete optimization heuristics with IOHprofiler. *Applied Soft Computing*, 88:106027, 2020.
- [10] Arkadiy Dushatskiy, Tanja Alderliesten, and Peter A. N. Bosman. A novel surrogate-assisted evolutionary algorithm applied to partition-based ensemble learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 583–591. Association for Computing Machinery, 2021.
- [11] Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: robust and efficient hyperparameter optimization at scale. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1437–1446. PMLR, February 2018.
- [12] Alexander I. J. Forrester, András Sóbester, and Andy J. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, December 2007. Publisher: Royal Society.
- [13] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing*, 380:20–35, March 2020. Publisher: Elsevier BV.
- [14] K. C. Giannakoglou, D. I. Papadimitriou, and I. C. Kampolis. Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6312–6329, September 2006.
- [15] Dirk Gorissen, Tom Dhaene, and Filip De Turck. Evolutionary model type selection for global surrogate modeling. *Journal of Machine Learning Research*, 10(71):2039–2078, 2009.
- [16] Dan Guo, Yaochu Jin, Jinliang Ding, and Tianyou Chai. Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Transactions on Cybernetics*, 49(3):1012–1025, 2019.
- [17] Gideon Hanse, Roy de Winter, Bas van Stein, and Thomas Bäck. Optimally weighted ensembles for efficient multi-objective optimization. In Giuseppe Nicosia, Varun Ojha, Emanuele La Malfa, Gabriele La Malfa, Giorgio Jansen, Panos M. Pardalos, Giovanni Giuffrida, and Renato Umerton, editors, *Machine Learning, Optimization, and Data Science*, pages 144–156, Cham, 2022. Springer International Publishing.
- [18] Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. Real-parameter black-box optimization benchmarking 2009: noiseless functions definitions. Research Report RR-6829, INRIA, 2009.
- [19] Naama Horesh, Thomas Bäck, and Ofer M. Shir. Predict or screen your expensive assay: DoE vs. surrogates in experimental combinatorial optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19*, pages 274–284, New York, NY, USA, 2019. Association for Computing Machinery. event-place: Prague, Czech Republic.
- [20] Qi Huang, Roy de Winter, Anna V. Kononova, Bas van Stein, and Thomas Bäck. Codes and supplementary materials of sama-diego, June 2022. <https://doi.org/10.5281/zenodo.6684442>.
- [21] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In Carlos A. Coello Coello, editor, *Learning and Intelligent Optimization*, pages 507–523. Springer Berlin Heidelberg, 2011.

- [22] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, June 2011.
- [23] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- [24] Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric P. Xing. ChemBO: Bayesian optimization of small organic molecules with synthesizable recommendations. Technical Report arXiv:1908.01425, arXiv, October 2019. arXiv:1908.01425 [physics, stat] type: article.
- [25] Fan Li, Xiwen Cai, and Liang Gao. Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems. *Applied Soft Computing*, 74:291–305, 2019.
- [26] Rui Li, Michael T. M. Emmerich, Jeroen Eggermont, Thomas Bäck, M. Schütz, J. Dijkstra, and J. H. C. Reiber. Mixed integer evolution strategies for parameter optimization. *Evol. Comput.*, 21(1):29–64, March 2013. Place: Cambridge, MA, USA Publisher: MIT Press.
- [27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, and et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.
- [28] Elena Raponi, Hao Wang, Mariusz Bujny, Simonetta Boria, and Carola Doerr. High dimensional bayesian optimization assisted by principal component analysis. In Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature – PPSN XVI*, pages 169–183, Cham, 2020. Springer International Publishing.
- [29] Frederik Rehbach, Martin Zaeferrer, Boris Naujoks, and Thomas Bartz-Beielstein. Expected improvement versus predicted value in surrogate-based optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 868–876. Association for Computing Machinery, 2020.
- [30] Abdelkader Sbihi and Richard W. Eglese. Combinatorial optimization and Green Logistics. *4OR*, 5(2):99–116, 2007.
- [31] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [32] Bas van Stein, Hao Wang, and Thomas Bäck. Automatic configuration of deep neural networks with parallel efficient global optimization. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2019.
- [33] Jörg Stork, Martina Friese, Martin Zaeferrer, Thomas Bartz-Beielstein, Andreas Fischbach, Beate Breiderhoff, Boris Naujoks, and Tea Tušar. Open issues in surrogate-assisted optimization. In Thomas Bartz-Beielstein, Bogdan Filipič, Peter Korošec, and El-Ghazali Talbi, editors, *High-Performance Simulation-Based Optimization*, pages 225–244. Springer International Publishing, Cham, 2020.
- [34] Tea Tušar, Dimo Brockhoff, and Nikolaus Hansen. Mixed-integer benchmark problems for single- and bi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19*, pages 718–726, New York, NY, USA, 2019. Association for Computing Machinery. event-place: Prague, Czech Republic.
- [35] Felipe A. C. Viana, Raphael T. Haftka, and Layne T. Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization*, 56(2):669–689, June 2013.
- [36] Hao Wang, Bas van Stein, Michael Emmerich, and Thomas Back. A new acquisition function for Bayesian optimization based on the moment-generating function. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 507–512, 2017.
- [37] Hu Wang, Fan Ye, Enying Li, and Guangyao Li. A comparative study of expected improvement-assisted global optimization with different surrogates. *Engineering Optimization*, 48(8):1432–1458, August 2016.
- [38] Martin Zaeferrer, Jörg Stork, Martina Friese, Andreas Fischbach, Boris Naujoks, and Thomas Bartz-Beielstein. Efficient Global Optimization for Combinatorial Problems. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14*, pages 871–878, New York, NY, USA, 2014. Association for Computing Machinery. event-place: Vancouver, BC, Canada.