Utilizing the Expected Gradient in Surrogate-assisted Evolutionary Algorithms

Kei Nishihara
Yokohama National University
Yokohama, Japan
nishihara-kei-jv@ynu.jp

Masaya Nakata
Yokohama National University
Yokohama, Japan
nakata-masaya-tb@ynu.ac.jp

ABSTRACT

In the field of surrogate-assisted evolutionary algorithms (SAEAs), Gaussian Process (GP) is a widely used technique to approximate the objective function. Although a GP model can provide an expected gradient of a function to be approximated, little attention has been paid to the utilization of the gradient information. Thus, this paper presents an expected gradient-based SAEA, in which the expected gradient of the objective function provided by the GP models is utilized to conduct an efficient local search. Specifically, the proposed algorithm first conducts a global search with a differential evolution algorithm to find promising regions of the search space. Then, it builds a GP model for each promising region, and a quasi-Newton method (L-BFGS-B) is executed on its model with guidance from the expected gradient. This gradient-based local search intends to sufficiently search the approximate objective function, by finding various local optimal solutions in an efficient manner. Experimental results show that our algorithm is competitive with state-of-the-art SAEAs on a single-objective optimization benchmark suite.

CCS CONCEPTS

• **Theory of computation → Continuous optimization**; *Bio-inspired optimization*; *Numeric approximation algorithms.*

KEYWORDS

surrogate-assisted evolutionary algorithm, expected gradient, quasi-Newton method, differential evolution, Gaussian process

ACM Reference Format:
Kei Nishihara and Masaya Nakata. 2023. Utilizing the Expected Gradient in Surrogate-assisted Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590694

1 INTRODUCTION

In many real-world applications, expensive optimization problems (EOPs) are often encountered, where fitness evaluations (FEs) are calculated from computationally expensive simulations [5]. This paper considers single-objective EOPs. Surrogate-assisted evolutionary algorithms (SAEAs) [4] are an effective approach to addressing EOPs. A basic idea of SAEAs is to estimate a promising solution among candidate ones by assessing their quality with surrogates. Usually, a surrogate is designed to approximate the objective function and is utilized to partially replace expensive FEs with its model predictions. Typically, Gaussian Process (GP) [8] and Radial Basis Function Network (RBFN) [10] are used as surrogate modeling techniques. Many modern approximation-based SAEAs are designed to alternately conduct global and local search phases with help of surrogates [14]. Roughly speaking, the global search phase aims to find new promising regions of the search space with a global model which approximates the entire landscape of the objective function. Whereas, the local search phase intends to intensively search a particularly promising region with a local approximation model adapted to its region.

For the local search phase, most existing SAEAs are designed to estimate promising solutions by optimizing the approximation model, i.e., the approximate objective function. EAs are usually employed to optimize it, and they are executed for each model update. Thus, the search capacity of EAs is crucial in determining the efficiency of the local search. However, many existing works set a small number of generations when executing EAs as the local search [2]. For instance, SAHO [9] executes Differential Evolution (DE) [12] with 30 generations in optimizing approximate objective functions. This probably intends 1) to reduce the runtime because the execution of EAs for each model update is time-consuming, and/or 2) to prevent solutions from being guided to the wrong region when approximation accuracy is low. However, the approximate objective function approximate model may not be sufficiently optimized with such a restricted number of generations.

Thus, a possible approach to address the above issue is to use gradient-based search algorithms for the local search if any gradient information is available. The gradient-based search is effective because the convergence speed can be enhanced and the convergence to the local optimum is guaranteed. Since GP is a stochastic process, the expected gradient of an approximate function at any point can be calculated [11]. This means that gradient-based search algorithms, such as quasi-Newton methods, can be applied in the optimization of the approximate objective function when the GP model is employed. Although the GP model is widely used in SAEAs, little attention has been paid to the utilization of the expected gradient. Moreover, while the expected gradient is often used in the literature on Bayesian Optimization field [3], the effectiveness of the ensemble with evolutionary algorithms is uncovered.

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590694

Accordingly, this paper presents an expected gradient-based SAEA, in which a DE and a quasi-Newton method are used for global and local phases, respectively. We use L-BFGS-B [15] as a representative quasi-Newton method. Specifically, the proposed algorithm first builds an RBFN to approximate the entire landscape of the objective function, and then DE is executed on its model to find promising regions of the search space. Next, a GP model is built for each promising region, and L-BFGS-B is applied multiple times to optimize each GP model with its expected gradient. This local search phase intends to sufficiently explore the approximate objective function by finding many local optimum solutions, by taking the advantage of the efficient gradient-based algorithm.

This paper is organized as follows. Section 2 introduces the GP model and its expected gradient. Section 3 explains the mechanism of our proposal. In Section 4, we compare our proposal with state-of-the-art SAEAs on CEC 2013 benchmark suite [6]. Finally, Section 5 describes our conclusion with future work.

2 GAUSSIAN PROCESS

For a scattered point sequence $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $f : \mathbb{R}^D \to \mathbb{R}$, the approximation of $f(\boldsymbol{x})$ is;

$$\hat{f}(\boldsymbol{x}) = \mu + \epsilon(\boldsymbol{x}), \tag{1}$$

where μ is a global regression model and corresponds to the mean of f, and $\epsilon(\boldsymbol{x})$ is the deviation from μ and is defined as $\mathcal{N}(0, \sigma^2)$. The correlation for the d-th dimensional deviation of any two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is expressed by the Gaussian correlation function as follows;

$$k_{ij,d}(x_{i,d}, x_{j,d}) = \exp(-\theta_d ||\boldsymbol{x}_{i,d} - \boldsymbol{x}_{j,d}||^2), \tag{2}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_D]^\mathsf{T}$ is a parameter that controls the strength of the correlation. Let the correlation function matrix K be an $n \times n$ matrix with $k_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{d=1}^n k_{ij,d}(x_{i,d}, x_{j,d})$ elements. The parameter $\boldsymbol{\theta}$ is searched in the range $\theta_d \in [\theta_d^l, \theta_d^u]$ and the value is used when the likelihood L is maximum. Then, μ and σ are given by;

$$L(\mu, \sigma, \boldsymbol{\theta}) = -\frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|K|), \tag{3}$$

$$\mu = \frac{\mathbf{1}^\mathsf{T} K^{-1} \boldsymbol{f}}{\mathbf{1}^\mathsf{T} K^{-1} \mathbf{1}},\ \sigma^2 = \frac{(\boldsymbol{f} - \mathbf{1}\mu)^\mathsf{T} K^{-1} (\boldsymbol{f} - \mathbf{1}\mu)}{n}. \tag{4}$$

Finally, Eq. (1) is deformed as follows;

$$\hat{f}(\boldsymbol{x}) = \mu + \boldsymbol{k}_{\boldsymbol{x}}^\mathsf{T} K^{-1} (\boldsymbol{f} - \mathbf{1}\mu), \tag{5}$$

where $\boldsymbol{k}_{\boldsymbol{x}}$ is the $n \times 1$ correlation vector for a new point $\boldsymbol{x}$ and each element $\boldsymbol{x} \in \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^n$, i.e., the scattered point sequence.

Considering GP is a stochastic process, the expected gradient of the objective function can be obtained. More precisely, the gradient is taken w.r.t. the new point $\boldsymbol{x}$ and the expectation is over the GP posterior distribution. Since the differentiation calculation is a linear operation, the expected gradient is equivalent to the gradient of the expected function value, i.e., the gradient of the approximate objective function, if the process is mean-square differentiable.

The differentiation of Eq. (5) is as follows;

$$g(\boldsymbol{x}) = J(\boldsymbol{x})^\mathsf{T} K^{-1} (\boldsymbol{f} - \mathbf{1}\mu), \tag{6}$$

$$J(\boldsymbol{x})_{i,d} = \frac{\partial k(x_{i,d}, x_d)}{\partial x_d}. \tag{7}$$

3 PROPOSAL

The proposed algorithm consists of the following three phases; the initialization, the DE-based global search with an RBFN model, and the expected gradient-based local search with GP models. We use the DACE model [8] as an implementation of GP model. The global search intends to roughly explore the search space, aiming to find promising regions to be searched by the local search. Whereas, the local search intends to estimate local optimum solutions distributed in multiple promising regions in an efficient manner, i.e., using a gradient-based algorithm (L-BFGS-B).

Algorithm 1 shows the procedure of our proposal, wherein the algorithm is terminated when the number of FEs, denoted as FE, reaches its maximum value $FE_{\max}$. In the proposed algorithm, all evaluated solutions with the original objective function are stored in an archive set $\mathcal{A}$. At the beginning of the search, N initial solutions $\{\boldsymbol{x}_i\}_{i=1}^N$ are produced with Latin Hypercube Sampling (LHS). Each initial solution is evaluated with the original objective function f. Then, all the initial solutions are inserted into $\mathcal{A}$ as $\mathcal{A} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^N$. As a main loop, the global and local search phases are executed alternately until the termination criterion is met, which are described below.

3.1 DE-based global search

The procedure of this phase is described in lines 5-17 of Algorithm 1. As aforementioned, an RBFN model, denoted as $\hat{f}_g$, is built with $\mathcal{A}$. Next, DE is executed to find a good solution on $\hat{f}_g$. For the DE settings, the top N solutions in $\mathcal{A}$, that is, solutions having the top N objective values, are used as the target solutions $\{\boldsymbol{x}_i\}_{i=1}^N$; and the number of generations is set to one. This is because the global search phase intends to find new promising regions; otherwise, DE is likely to find solutions belonging to regions that have been already searched in the local search phase. Such a "one-shot" exploration strategy has been widely used in SAEAs [1, 13]. Accordingly, new N offspring solutions $\{\boldsymbol{u}_i\}_{i=1}^N$ are generated in line 9 with the DE procedure. We use the *best/1* mutation and the *binomial* crossover strategies. The approximate objective value for each offspring solution $\hat{f}_g(\boldsymbol{u}_i)$ is calculated. Finally, the offspring solution having the minimum value of $\hat{f}_g(\boldsymbol{u}_i)$ is evaluated with the original objective function, and it is inserted into $\mathcal{A}$.

3.2 Expected gradient-based local search

The procedure of this phase is described in lines 19-31 of Algorithm 1. To begin with, the algorithm estimates multiple promising regions of the search space. Specifically, the algorithm produces N samples $\{\boldsymbol{s}_i\}_{i=1}^N$, which are randomly distributed in the whole search space; LHS is used to produce these samples. The approximate objective values of all the samples are calculated by $\hat{f}_g$, i.e., the RBFN model constructed in the global search phase; and M *superior* samples $\{\boldsymbol{s}_m^*\}_{m=1}^M$ are identified as having the top M values of $\hat{f}_g(\boldsymbol{s})$. The proposed algorithm then estimates a promising region that exists close to each superior sample $\boldsymbol{s}_m^*$.

Next, the proposed algorithm builds M local GP models, in which each model is specialized to a region close to its corresponding superior sample; and then the algorithm conducts multiple runs of L-BFGS-B on each GP model. Specifically, for each superior sample $\boldsymbol{s}_m^*$,

Algorithm 1 Proposed Algorithm

Initialize $\{\boldsymbol{x}_i\}_{i=1}^N$ by LHS and Evaluate
$\mathcal{A} = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}_{i=1}^N$, $FE = N$
while $FE < FE_{\max}$ **do**
  // Global Exploration Phase //
  Build the global RBFN surrogate model $\hat{f}_g$ with $\mathcal{A}$
  Select top N solutions $\{\boldsymbol{x}_i\}_{i=1}^N$ and their fitness values from $\mathcal{A}$
  for $i = 1$ **to** N **do**
    $\boldsymbol{v}_i \leftarrow$ *best/1-mutation* with $\{\boldsymbol{x}_i\}_{i=1}^N$
    $\boldsymbol{u}_i \leftarrow$ *binomial-crossover* with $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$
    $\hat{f}_g(\boldsymbol{u}_i) \leftarrow$ Evaluate $\boldsymbol{u}_i$ with $\hat{f}_g$
  end for
  $\boldsymbol{u}^* = \arg\min \hat{f}_g(\boldsymbol{u}_i)$
  $f(\boldsymbol{u}^*) \leftarrow$ Evaluate $\boldsymbol{u}^*$ with the original objective function f
  $FE = FE + 1$, $\mathcal{A} = \mathcal{A} \cup \{(\boldsymbol{u}^*, f(\boldsymbol{u}^*))\}$
  if $FE \geq FE_{\max}$ **then**
    return the best solution in $\mathcal{A}$
  end if
  // Local Exploitation Phase //
  Generate N samples $\{\boldsymbol{s}_i\}_{i=1}^N$ by LHS and Evaluate with $\hat{f}_g$
  Select M *superior* samples $\{\boldsymbol{s}_m^*\}_{m=1}^M$ having top M values $\hat{f}_g(\boldsymbol{s})$
  Set the candidate set $C = \emptyset$
  for $m = 1$ **to** M **do**
    $\mathcal{D}_m \leftarrow K$ closest solutions to $\boldsymbol{s}_m^*$ from $\mathcal{A}$
    $\hat{f}_m \leftarrow$ Build the GP surrogate model with $\mathcal{D}_m$
    Generate L initial points by LHS with the range $[l_{m,j}, u_{m,j}]^D$
    Conduct L-BFGS-B optimization for each initial points with $\hat{f}_m$
    Obtain L candidate solutions and Add them and their $\hat{f}_m$ to C
  end for
  $\boldsymbol{x}^* \leftarrow$ the solution having the best $\hat{f}_m(\boldsymbol{x}), m = \{1, 2, \ldots, M\}$
  $f(\boldsymbol{x}^*) \leftarrow$ Evaluate $\boldsymbol{x}^*$ with the original function f
  $FE = FE + 1$, $\mathcal{A} = \mathcal{A} \cup \{(\boldsymbol{x}^*, f(\boldsymbol{x}^*))\}$
end while
return the best solution in $\mathcal{A}$

the K closest solutions to $\boldsymbol{s}_m^*$ in Euclidean distance are selected from $\mathcal{A}$ and used as training samples, forming a training dataset $\mathcal{D}_m$. The local GP model $\hat{f}_m$ is then built with these training samples. Next, to conduct multiple runs of L-BFGS-B, L different initial points are randomly generated, wherein each point is sampled from a region close to $\boldsymbol{s}_m^*$, that is, $[l_{m,j}, u_{m,j}]^D$ with $l_{m,j} = \min_{\boldsymbol{x} \in \mathcal{D}_m} x_j$ and $u_{m,j} = \max_{\boldsymbol{x} \in \mathcal{D}_m} x_j$. Next, L-BFGS-B is executed for L times with these different initial points. Consequently, L candidate solutions are obtained. Finally, those candidate solutions along with their approximate objective values of $\hat{f}_m$ are stored in a candidate set C.

After the execution of the above procedure, the candidate set C includes $L \times M$ candidate solutions, which are likely local optimal solutions for the estimated promising regions. Note that L solutions are obtained from M GP models. The proposed algorithm compares the quality of these solutions and identifies the most promising solution to be evaluated with the original objective function. Specifically, the algorithm selects and then evaluates the candidate solution having the best approximate objective value among the ones in C.

4 EXPERIMENT

This section compares the performances of the proposed algorithm and the following five state-of-the-art SAEAs; S-JADE [2], SAHO [9], GPEME [7], IKAEA [13], and GSGA [1]. GPEME and IKAEA are representative GP-based SAEAs while S-JADE and SAHO are RBFN-based SAEAs. GSGA uses both the GP model and the RBFN model. Similar to the proposed algorithm, S-JADE and GSGA alternately conduct the global and local search phases.

4.1 Experimental Design

We use the IEEE CEC 2013 real-parameter single-objective benchmark function suite [6]. The benchmark suite consists of five unimodal functions F1-F5, 15 multi-modal functions F6-F20, and eight composition functions F21-F28. The bound constraint is $[-100, 100]^D$ for all the functions. We set the problem dimension as $D \in \{10, 30\}$.

For hyper-parameters of the compared algorithms, the same settings are used as in their original, which are summarized in Table 1. For our proposal, $N = 100, F = 0.5, CR = 0.9, M = 3, K = 50$, $L = 5D$, RBFN kernel = cubic, $\theta \in [10^{-5}, 10^2]$, and $\theta_{init} = 10^{-2}$.

The maximum number of FEs FE_{max} is set to 1,000.The performance of each algorithm is evaluated as the best objective value and is reported as the mean value of 15 independent runs. The Wilcoxon rank-sum test is used to find a significant difference between their performances with the significance level of 0.05.

4.2 Result

Tables 2 and 3 show the best objective values discovered at 1,000 FEs for $D = 10$ and 30, respectively. Note that the best and the worst values among algorithms are highlighted in green bold and pink, respectively. The symbols "+" and "−" indicate that the performance of proposed algorithm is significantly worse and better than that of a compared algorithm, respectively; "∼" indicates that there is no significant difference between their performances.

From Table 2, the performance of the proposed algorithm is competitive to compared algorithms. The effectiveness of our proposed algorithm can be further observed for $D = 30$ (see Table 3). In addition, the proposed algorithm tends to perform well on multi-modal functions (F6-F28). This result empirically supports the effectiveness of our local search algorithm; the proposed algorithm conducts the gradient-based search with different initial points to obtain various local optima simultaneously.

Table 4 reports the summary of statistical results, i.e., "+/−/∼", when FE_{max} is set to 200, 400, ..., 1,000. From the table, S-JADE and SAHO perform well ad 400 FE, where the proposed algorithm may struggle to find promising regions during the global phases. However, the proposed algorithm sufficiently outperforms all the algorithms when $FE \leq 600$.

5 CONCLUSION

This paper proposed an expected gradient-based SAEA, which utilizes the expected gradient of GP models for the local search. A main advantage of the proposed algorithm is that the GP model, that is, the approximate objective function can be intensively optimized with the efficient gradient-based local search. Experimental result showed that the proposed algorithm is very competitive with the state-of-the-art SAEAs.

In future work, we will integrate different gradient-based algorithms into our proposed framework and validate their effectiveness. We will also extend our approach to multi-objective optimization problems.

Table 1: Hyper-parameter settings.

Algorithm	Hyper-parameters
S-JADE	$N = 30$, $F_{out} = 0.5$, $CR_{out} = 0.75$, $p_{pbest_{out}} = 0.05$, $F_{in} = 0.5$, $CR_{out} = 0.5$, $p_{pbest_{in}} = 0.1$, $std_F = 0.1$, $std_{CR} = 0.1$, $L = 10$, $\epsilon = 0.01$, $c = 0.1$, $FE_{max_{in}} = 2,000$, kernel = cubic, and $r = rand(0,\ 1.25)$.
SAHO	$N = 5D (D < 50)$ or $100 + \lfloor D/10 \rfloor (D \geq 50)$, $F = 0.9$, $CR = 0.5$, $K = 30$, $neighbor = 5D (D < 50)$ or $D (D \geq 50)$, and kernel = cubic.
GPEME	$N = 100, F = 0.8, CR = 0.8, \tau = 100, \lambda = 50, l = 4, \omega = 2$, regression = zero-order, correlation = Gaussian, $\theta \in [10^{-5}, 10^2]$, and $\theta_{init} = 10^{-2}$.
IKAEA	$N_{init} = 100$, $N = 50$, $F = 0.8$, $CR = 0.8$, $\epsilon = 10^{-6}$, $[\theta_d^l, \theta_d^u] = [10^{-5}, 10^2]$, $\theta_{init} = 1$, $\ln L_{\min} = -10^8$.
GSGA	$N = 50$, $P_{cross} = 0.9$, $P_{mut} = 0.1$, $N_s = TS = 10$, $k_{\max} = 3$, $MP = 0.5$, $pc = 0.1$, $d_{limit} = 0.01$, $es = \{1, 2\}$, $\rho = \{0.25, 0.75\}$, kernel = cubic, $\theta \in [0.01, 20]$, and $\theta_{init} = 10$.

Table 2: The best fitness values discovered at $1,000$ fitness evaluations for $D = 10$.

	S-JADE	SAHO	GPEME	IKAEA	GSGA	Proposal
F1	7.80E-06 −	**2.88E-28 +**	5.56E-12 +	4.32E-02 −	1.18E-05 −	3.96E-10
F2	2.71E+06 −	1.83E+06 −	1.36E+07 −	**8.49E+04 +**	5.94E+06 −	7.09E+05
F3	6.09E+09 +	5.57E+10 −	7.46E+09 ∼	4.93E+09 +	**4.14E+09 +**	8.73E+09
F4	1.94E+04 −	2.10E+04 −	5.04E+04 −	1.58E+04 −	3.68E+04 −	**1.06E+04**
F5	4.35E+01 +	4.13E-02 +	**1.57E-03 +**	2.12E+02 ∼	1.07E+02 +	2.52E+02
F6	7.89E+00 +	9.18E+00 +	8.94E+00 +	**3.06E+00 +**	1.08E+01 ∼	1.41E+01
F7	1.14E+02 ∼	3.20E+02 −	1.26E+02 ∼	1.12E+02 +	**7.98E+01 +**	1.45E+02
F8	**2.07E+01** ∼	2.08E+01 ∼	2.08E+01 ∼	**2.07E+01** ∼	**2.07E+01** ∼	**2.07E+01**
F9	6.73E+00 ∼	7.00E+00 ∼	**5.33E+00** ∼	5.83E+00 ∼	8.16E+00 −	6.42E+00
F10	4.24E-01 −	6.01E-01 −	9.81E-01 −	4.86E-01 −	4.66E+00 −	**1.09E-01**
F11	3.93E+01 −	4.89E+01 −	**1.67E+01** ∼	8.20E+01 −	2.54E+01 ∼	2.12E+01
F12	5.15E+01 −	3.81E+01 −	3.78E+01 −	8.64E+01 −	3.09E+01 −	**2.26E+01**
F13	5.64E+01 ∼	5.77E+01 ∼	4.76E+01 ∼	7.78E+01 −	6.15E+01 −	**4.66E+01**
F14	1.77E+03 −	1.27E+03 −	9.17E+02 ∼	**8.39E+02** ∼	1.01E+03 ∼	9.35E+02
F15	1.95E+03 −	1.62E+03 −	2.02E+03 −	1.49E+03 ∼	1.45E+03 ∼	**1.32E+03**
F16	2.50E+00 ∼	2.25E+00 ∼	2.32E+00 ∼	2.27E+00 ∼	**2.11E+00** ∼	2.30E+00
F17	5.03E+01 −	3.05E+01 ∼	2.83E+01 −	8.65E+01 −	4.77E+01 −	**2.25E+01**
F18	5.66E+01 −	3.62E+01 ∼	5.97E+01 −	8.66E+01 −	6.58E+01 −	**3.10E+01**
F19	1.01E+01 +	2.65E+02 +	1.04E+02 +	3.15E+02 +	**4.29E+00 +**	2.50E+03
F20	4.38E+00 −	4.48E+00 −	**3.92E+00** ∼	4.10E+00 ∼	4.27E+00 ∼	4.09E+00
F21	4.26E+02 +	4.50E+02 +	3.87E+02 +	6.67E+02 ∼	**3.80E+02 +**	5.15E+02
F22	1.88E+03 −	1.69E+03 −	**9.89E+02 +**	1.11E+03 ∼	1.22E+03 ∼	1.28E+03
F23	2.26E+03 −	1.73E+03 −	1.85E+03 −	1.63E+03 −	1.61E+03 −	**1.27E+03**
F24	2.18E+02 ∼	2.20E+02 ∼	**2.16E+02** ∼	2.27E+02 −	2.22E+02 ∼	2.20E+02
F25	2.17E+02 ∼	**2.16E+02** ∼	**2.16E+02 +**	2.28E+02 −	2.22E+02 ∼	2.20E+02
F26	2.07E+02 ∼	2.07E+02 −	2.32E+02 −	2.08E+02 −	**1.85E+02 +**	1.98E+02
F27	**4.51E+02 +**	5.09E+02 ∼	5.07E+02 ∼	6.09E+02 −	5.09E+02 ∼	5.30E+02
F28	8.34E+02 +	1.27E+03 ∼	**3.02E+02 +**	9.87E+02 ∼	3.98E+02 +	1.10E+03
+/−/∼	7/13/8	5/13/10	8/9/11	5/13/10	7/10/11	

NOTE: Signs +, −, and ∼ indicate that the proposal significantly underperforms, outperforms, is comparable with an algorithm, respectively.

Table 3: The best fitness values discovered at $1,000$ fitness evaluations for $D = 30$.

	S-JADE	SAHO	GPEME	IKAEA	GSGA	Proposal
F1	6.92E+00 +	**1.88E-15 +**	6.71E+02 −	3.12E-02 +	3.48E-04 +	2.75E+02
F2	9.40E+07 −	**1.06E+07 +**	1.41E+08 −	7.57E+07 −	1.05E+08 −	3.61E+07
F3	2.07E+15 ∼	4.05E+17 −	4.59E+11 ∼	1.81E+16 ∼	**2.95E+11 +**	5.69E+13
F4	**8.40E+04 +**	1.25E+05 ∼	1.75E+05 −	1.06E+05 ∼	1.61E+05 −	1.17E+05
F5	3.12E+03 ∼	**1.79E+02 +**	1.34E+03 +	3.07E+03 ∼	2.75E+03 ∼	2.53E+03
F6	1.08E+02 ∼	**4.22E+01 +**	7.66E+01 +	2.02E+02 −	1.05E+02 ∼	1.28E+02
F7	2.06E+04 −	2.09E+05 −	1.13E+03 ∼	1.11E+05 ∼	**4.49E+02 +**	2.61E+03
F8	**2.12E+01** ∼	**2.12E+01** ∼	**2.12E+01** ∼	**2.12E+01** ∼	**2.12E+01** ∼	**2.12E+01**
F9	3.75E+01 −	2.97E+01 ∼	**2.85E+01** ∼	4.40E+01 −	3.87E+01 −	2.96E+01
F10	5.84E+01 ∼	**1.25E+00 +**	2.98E+02 −	9.39E+00 +	1.22E+02 −	6.44E+01
F11	2.87E+02 −	2.80E+02 −	1.69E+02 −	2.97E+02 −	2.52E+02 −	**1.24E+02**
F12	3.02E+02 −	2.39E+02 −	2.94E+02 −	3.00E+02 −	2.87E+02 −	**1.38E+02**
F13	3.18E+02 −	3.00E+02 −	2.98E+02 −	2.96E+02 −	3.33E+02 −	**2.58E+02**
F14	7.90E+03 −	6.14E+03 ∼	5.48E+03 ∼	6.36E+03 ∼	7.05E+03 −	**5.30E+03**
F15	8.67E+03 −	**6.65E+03** ∼	8.90E+03 −	8.80E+03 −	8.62E+03 −	7.11E+03
F16	4.51E+00 ∼	4.59E+00 ∼	4.46E+00 ∼	4.74E+00 ∼	4.58E+00 ∼	**4.40E+00**
F17	2.74E+02 ∼	2.70E+02 ∼	2.56E+02 ∼	3.14E+02 −	2.85E+02 −	**2.44E+02**
F18	**2.91E+02 +**	2.92E+02 ∼	3.28E+02 ∼	3.24E+02 ∼	3.44E+02 ∼	3.21E+02
F19	4.67E+04 ∼	2.95E+05 −	7.49E+03 +	8.21E+03 +	**1.88E+02 +**	4.39E+04
F20	1.50E+01 ∼	1.50E+01 ∼	**1.48E+01** ∼	1.50E+01 −	1.50E+01 −	1.49E+01
F21	2.41E+03 +	4.34E+03 −	4.66E+03 −	2.43E+03 +	**1.56E+03 +**	2.75E+03
F22	8.47E+03 −	6.62E+03 ∼	5.90E+03 ∼	6.74E+03 ∼	7.55E+03 −	**5.68E+03**
F23	9.17E+03 −	**6.42E+03 +**	9.28E+03 −	9.34E+03 −	9.06E+03 −	7.66E+03
F24	2.99E+02 −	2.88E+02 ∼	**2.72E+02 +**	2.99E+02 −	3.03E+02 −	2.84E+02
F25	3.16E+02 −	3.02E+02 −	**2.84E+02 +**	3.34E+02 −	3.08E+02 −	2.93E+02
F26	**3.35E+02** ∼	3.59E+02 ∼	3.85E+02 −	3.58E+02 ∼	3.64E+02 ∼	3.50E+02
F27	1.17E+03 −	1.08E+03 ∼	**1.03E+03** ∼	1.49E+03 −	1.28E+03 −	1.08E+03
F28	4.65E+03 ∼	7.51E+03 −	5.38E+03 −	5.37E+03 ∼	**4.03E+03** ∼	4.16E+03
+/−/∼	4/13/11	6/9/13	5/12/11	4/13/11	5/16/7	

Table 4: Significant differences regarding findings for "+/−/∼" between our proposal and state-of-the-art SAEAs.

D	FE	vs S-JADE	vs SAHO	vs GPEME	vs IKAEA	vs GSGA
10	200	11/ 1/16	12/ 0/16	2/12/14	5/10/13	6/ 5/17
	400	7/ 8/13	9/ 2/17	5/11/12	4/13/11	7/12/ 9
	600	8/11/ 9	6/10/12	7/11/10	4/11/13	7/15/ 6
	800	7/13/ 8	5/13/10	5/11/12	4/12/12	7/13/ 8
	1,000	7/13/ 8	5/13/10	8/ 9/11	5/13/10	7/10/11
30	200	12/ 1/15	4/ 6/18	0/14/14	2/15/11	6/ 5/17
	400	8/ 4/16	9/ 6/13	3/ 9/16	2/ 8/18	4/10/14
	600	6/ 7/15	8/ 7/13	4/ 8/16	4/ 7/17	5/12/11
	800	7/11/10	6/10/12	4/10/14	5/11/12	6/14/ 8
	1,000	4/13/11	6/ 9/13	5/12/11	4/13/11	5/16/ 7

REFERENCES

[1] Xiwen Cai, Liang Gao, and Xinyu Li. 2020. Efficient Generalized Surrogate-Assisted Evolutionary Algorithm for High-Dimensional Expensive Problems. *IEEE Trans. Evol. Comput.* 24, 2 (April 2020), 365–379.

[2] Xiwen Cai, Liang Gao, Xinyu Li, and Haobo Qiu. 2019. Surrogate-guided differential evolution algorithm for high dimensional expensive problems. *Swarm Evol. Comput.* 48 (Aug. 2019), 288–311.

[3] David Eriksson, Michael Pearce, Jacob R Gardner, Ryan Turner, and Matthias Poloczek. 2019. Scalable global optimization via local Bayesian optimization. In *Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*. Curran Associates Inc., Red Hook, NY, USA, 5496–5507.

[4] Yaochu Jin, Handing Wang, Tinkle Chugh, Dan Guo, and Kaisa Miettinen. 2019. Data-Driven Evolutionary Optimization: An Overview and Case Studies. *IEEE Trans. Evol. Comput.* 23, 3 (June 2019), 442–458.

[5] Jian-Yu Li, Zhi-Hui Zhan, and Jun Zhang. 2022. Evolutionary Computation for Expensive Optimization: A Survey. *Mach, Intell. Res.* 19, 1 (Feb. 2022), 3–23.

[6] J J Liang, B Y Qu, P N Suganthan, and Alfredo G Hernández-Díaz. 2013. *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization.* Technical Report. Comput. Intell. Lab., Zhengzhou Univ., China, and Nanyang Tech. Univ., Singapore. 281–295 pages.

[7] Bo Liu, Qingfu Zhang, and Georges G E Gielen. 2014. A Gaussian Process Surrogate Model Assisted Evolutionary Algorithm for Medium Scale Expensive Optimization Problems. *IEEE Trans. Evol. Comput.* 18, 2 (April 2014), 180–192.

[8] S N Lophaven, H B Nielsen, and J Søndergaard. 2002. *DACE: A MATLAB Kriging Toolbox.* Technical Report IMM-REP-2002-12. Informatics and Mathematical Modelling, DTU.

[9] Jeng-Shyang Pan, Nengxian Liu, Shu-Chuan Chu, and Taotao Lai. 2021. An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems. *Inf. Sci.* 561 (June 2021), 304–325.

[10] J Park and I W Sandberg. 1991. Universal approximation using radial-basis-function networks. *Neural Comput.* 3, 2 (1991), 246–257.

[11] Carl Edward Rasmussen and Christopher K I Williams. 2005. *Gaussian Processes for Machine Learning.* MIT Press.

[12] Rainer Storn and Kenneth Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Global Optimiz.* 11 (1997), 341–359.

[13] Dawei Zhan and Huanlai Xing. 2021. A Fast Kriging-Assisted Evolutionary Algorithm Based on Incremental Learning. *IEEE Trans. Evol. Comput.* 25, 5 (Oct. 2021), 941–955.

[14] Liang Zheng, Jinyue Shi, and Youpeng Yang. 2023. A two-stage surrogate-assisted meta-heuristic algorithm for high-dimensional expensive problems. *Soft Computing* (Feb. 2023).

[15] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.* 23, 4 (Dec. 1997), 550–560.