



A multi-model assisted differential evolution algorithm for computationally expensive optimization problems

Haibo Yu¹ · Li Kang² · Ying Tan³ · Jianchao Zeng¹ · Chaoli Sun³

Received: 25 January 2021 / Accepted: 1 June 2021 / Published online: 11 June 2021
© The Author(s) 2021

Abstract

Surrogate models are commonly used to reduce the number of required expensive fitness evaluations in optimizing computationally expensive problems. Although many competitive surrogate-assisted evolutionary algorithms have been proposed, it remains a challenging issue to develop an effective model management strategy to address problems with different landscape features under a limited computational budget. This paper adopts a coarse-to-fine evaluation scheme basing on two surrogate models, i.e., a coarse Gaussian process and a fine radial basis function, for assisting a differential evolution algorithm to solve computationally expensive optimization problems. The coarse Gaussian process model is meant to capture the general contour of the fitness landscape to estimate the fitness and its degree of uncertainty. A surrogate-assisted environmental selection strategy is then developed according to the non-dominance relationship between approximated fitness and estimated uncertainty. Meanwhile, the fine radial basis function model aims to learn the details of the local fitness landscape to refine the approximation quality of the new parent population and find the local optima for real-evaluations. The performance and scalability of the proposed method are extensively evaluated on two sets of widely used benchmark problems. Experimental results show that the proposed method can outperform several state-of-the-art algorithms within a limited computational budget.

Keywords Differential evolution · Surrogate · Gaussian process · Radial basis function · Multi-model management · Expensive problem

Introduction

The differential evolution (DE) algorithm [50] has been shown to be very powerful in solving discontinuous, non-differential, multi-modal, and not well-defined problems. As a population-based search metaheuristic, however, DE is also subject to challenges when solving computationally expensive problems [5,34,36,69,71]. Thus, computationally cheap surrogate models, including polynomial regression (PR) [34,64], radial basis function (RBF) [22,32,53,59], artificial neural network (ANN) [14,23,27], support vector machines (SVM) [63,68], and Kriging or Gaussian process (GP) [3,6,16,41,42] are often used as a substitution of the original computationally expensive objective function to reduce the required computational cost of evolutionary optimization [1,8,29].

While early research on surrogate-assisted evolutionary optimization typically used single surrogate models, either in evolutionary search [26,27] or in local search [46], multi-model assisted evolutionary algorithms have become

✉ Haibo Yu
tyustyuhaibo@126.com

✉ Chaoli Sun
chaoli.sun.cn@gmail.com

Li Kang
kl199047@163.com

Ying Tan
tanying1965@gmail.com

Jianchao Zeng
zengjianchao@263.net

¹ Institute of Big Data and Visual Computing, North University of China, Taiyuan 030051, China

² Key Laboratory of Integrated Regulation and Resource Development on Shallow Lakes, Ministry of Education, Hohai University, Nanjing 210098, China

³ Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

increasingly popular for different reasons. A straightforward motivation of using multiple surrogates is to enhance the approximation accuracy and predict the reliability of the approximated fitness using a neural network ensemble [28], or to promote the diversity of search in both single- and multi-objective optimization [35,39,73]. Tang et al. [61] presented a hybrid surrogate assisted particle swarm optimization, in which an RBF model constructed by interpolating the residual errors was added to a low order polynomial regression model to form the final hybrid surrogate model to approximate the original fitness landscape. A selective ensemble was proposed for offline optimization [66] to adapt the surrogate where the on-line update of the models is impossible. A more sophisticated idea is to use multiple surrogates to approximate the global and local features of the fitness function, respectively. For example, Zhou et al. [74] suggested to combine both the global and local surrogate models to accelerate the evolutionary optimization, in which a prescreen procedure was presented based on a global Gaussian process model and an RBF-based trust-region search strategy was proposed in the spirit of Lamarckian learning. Tenne et al. [62] developed an improved version of hierarchical surrogate assisted memetic algorithm by adaptively switching global and local RBF models based on the leave-one-out cross-validation errors. Sun et al. [59] suggested to use a global surrogate model to smooth out the local optima and a local surrogate model was trained for each individual to approximate the fitness as accurately as possible. The authors further proposed a cooperative swarm algorithm for high-dimensional computationally expensive problems [58], in which an RBF network was utilized to smooth out the local optima and assist the social learning particle swarm optimization to quickly locate the region where the global optimum is located. The global RBF is combined with a local fitness estimation strategy [60] to improve the accuracy of the approximated fitness. In the context of off-line data-driven optimization, where no fitness evaluations are allowed, a low-order polynomial has been used to generate training data for online update of a Gaussian process [9]. A slightly different approach was reported in Isaacs et al. [24] that employed multiple RBF models for multiple database subsets obtained by k-means clustering to present a spatially distributed surrogates assisted evolutionary algorithm for multi-objective optimization. The third line of research on multi-model assisted evolutionary optimization aims to exploit the uncertainty information multiple surrogates can offer, similar to the infill criteria in Bayesian optimization [57]. Wang et al. [65] proposed an ensemble-based model management method for surrogate assisted particle swarm optimization which searches for the promising and most uncertain candidate solutions to be evaluated using the expensive simulation model. In [20], a heterogeneous ensemble is employed to replace the Gaussian process for estimating the mean as well as the standard

deviation of the fitness so that infill criterion driven model management strategies can be applied to surrogate-assisted optimization of high-dimensional problems. Note that the computational complexity of the Gaussian process increases in cubic with the number of training data, making it unrealistic for the high-dimensional problems.

Surrogate-assisted DE algorithms for solving computationally expensive problems have also been widely studied. Mallipeddi et al. [44] proposed to keep generating candidate offspring for each parent solution until the fitness of the candidate offspring approximated by the Kriging model is better than that of its parent or the number of generated candidate offspring reaches the predefined maximum. Then the candidate offspring with the best-approximated fitness will be evaluated using the expensive objective function and compared with its parent. The winner will be passed to the next generation. Similarly, Gong et al. [19] proposed to generate multiple offspring candidates for each parent by applying different reproduction operators, and the one with the maximum density value estimated by the Parzen window method is chosen to be the offspring to be evaluated using the expensive objective function. After that, each offspring will compete with its parent to survive. To further reduce the number of required real fitness evaluations, Vincenzi et al. [64] proposed an infill sampling strategy that is the weighted sum of the estimated fitness and the distance to the history sample points to assist a DE algorithm. Liu et al. [41] adopted the lower confidence bound for a Gaussian process assisted DE algorithm for solving medium-scale expensive optimization problems. Note, however, that the Sammon mapping technique was used for reducing the original decision space to a low-dimensional space. Multi-model assisted DE algorithms have also been proposed for solving expensive optimization problems [43]. A soft-margin support vector classifier was employed to divide a population combining the parent and offspring individuals into two groups according to their estimated fitness. The group of individuals having better fitness were re-evaluated using a regression model to determine which ones will be eventually evaluated using the real objective function. In [15], Elsayed et al. proposed to use a global Kriging model to approximate the fitness of individuals of a DE algorithm, then the expected improvement (EI) criterion is used for selecting solutions to be evaluated using the real objective function. Liu et al. [40] proposed a surrogate-assisted DE for multi-fidelity evolutionary optimization of computationally expensive problems.

While many surrogate-assisted evolutionary algorithms incorporate the uncertainty of the estimated fitness in model management, the uncertainty propagation in environmental selection lacks consideration. As a result, the individuals selected as the parents of the next generation may be located very far away from the optimum and the recently evaluated individuals, making the evolutionary algorithm more likely

to get stuck in a minimum of the surrogate model. To avoid premature convergence and promote the diversity of the population, this work proposes a selection strategy that considers both the approximated fitness values and the estimated degree of uncertainty. To this end, a GP surrogate model is adopted to estimate the fitness value and the uncertainty of the estimated fitness, which are used as two criteria in selecting parents. Different from most existing GP-assisted evolutionary algorithms, this work employs a second surrogate, which is an RBF model, for determining which offspring individuals are to be evaluated using the real fitness function. The main reason is that the computational complexity of the GP is cubic to the number of training data, which makes the GP unsuited for accurately modeling the local fitness landscape in a high-dimensional space. By contrast, RBF models have shown to be computationally efficient for local modeling of both low- and high-dimensional problems [25]. Due to the above reasons, we introduce an RBF model to approximate the local details of the fitness landscape and employ it to refine the approximation quality of the new parent population prevailing in the environmental selection, apart from the GP as a global surrogate for selecting parents. To promote the exploitation, a local search is applied to find the optimum of the RBF model in the local area covered by the selected parents and the top-ranking evaluated samples for fitness evaluation using the expensive real objective function. We term the overall algorithm multi-model assisted differential evolution algorithm, MADE for short.

The main contributions of this paper can be summarized as follows:

- (1) A coarse-to-fine evaluation scheme basing on two surrogate models, i.e., a coarse Gaussian process and a fine radial basis function, is proposed for assisting the differential evolution algorithm to solve computationally expensive optimization problems.
- (2) A surrogate-assisted environmental selection strategy is developed to select promising parents for the next generation according to the non-dominance relationship between approximated fitness and estimated uncertainty.
- (3) In each generation, a fine surrogate model is used to refine the approximation quality of the selected parent individuals from the environmental selection and screen out promising individuals for real-evaluation. On the other hand, a surrogate-based local optimization is performed according to the improvement of the current best to strike a balance between exploration and exploitation.
- (4) Systematic experiments have been conducted to validate the effectiveness and efficiency of MADE and compare it with several state-of-the-art algorithms on symmetric fitness landscapes with optimum lying on the origin and non-symmetric fitness landscapes with non-origin optimum.

The remainder of this paper is divided into the following sections. Section 2 describes in detail the proposed multi-model assisted differential evolution algorithm. Section 3 empirically assesses the performance of the MADE algorithm on two sets of benchmark problems with different types of fitness landscapes. Finally, Sect. 4 concludes the paper with a summary of this work and suggestions for future research.

Multi-model assisted DE

In canonical surrogate-assisted differential evolution algorithm, parent individuals for the next generation are normally selected according only to the approximated fitness, while overlooking the estimated uncertainty of the approximated fitness. Consequently, surrogate-assisted evolutionary algorithms are very likely to converge prematurely [3], when the surrogate fails to correctly capture the landscape of the fitness function and has a large uncertainty in fitness approximation. Therefore, we propose to select also the individuals whose approximated fitness has a large degree of uncertainty. In addition, a fine surrogate model is constructed to refine the approximation quality of these individuals and select the best one as well as the local optimum in the neighborhood of these individuals for fitness evaluation using the real computationally expensive function.

Figure 1 gives a diagram of the proposed algorithm. From Fig. 1, we can see that two surrogate models, a coarse GP model and a fine RBF model, are introduced to assist the optimization. The coarse GP model [54] is mainly for selecting parent individuals for the next generation in the environmental selection, while the fine RBF model, which has been shown effective for high-dimensional problems [72], is adopted for two purposes. One is to refine the approximation quality of the selected parent individuals from the GP-assisted environmental selection and choose the best one for fitness evaluation using the real computationally expensive objective function, and the other is to search the local optimum for real-evaluation from the region covered by the new parent population and the best-evaluated samples.

Before the optimization starts, a number of solutions are sampled using the symmetric Latin hypercube design (SLHD) method [33] and they are evaluated using the computationally expensive objective function. These offline sampled solutions as well as their fitness values are stored in an archive. The best solutions in the archive are used to seed the initial population of the DE, and then the crossover and mutation operators of DE are conducted to generate the trial/offspring candidate solutions. After that, a coarse GP model is first constructed using the collected solutions from the archive that are in the vicinity of the parents and the trial/offspring population. A parent for the next generation will then be selected according to the dominance relationship

based on the approximated fitness and the estimated uncertainty. Once all parents for the next generation are selected, a fine RBF model is then built using adjacent real-evaluated solutions of the selected parents to refine the approximation quality of the parent individuals, and the best one among these selected parents is chosen for real-evaluation if it is superior to the current global best. Otherwise, an RBF-assisted local optimization will be implemented to found a local optimum for fitness evaluation using the real computationally expensive objective function. Afterwards, the archive and the global best are updated by the newly evaluated solutions. And finally the above process is repeated until the termination condition is reached.

Algorithm 1 gives the pseudo-code of the MADE algorithm. The proposed algorithm follows the main steps of the basic differential evolution algorithm, including offline solution sampling, population initialization, generation of offspring using mutation and crossover, and selection of parents. The main difference between the MADE algorithm and the canonical differential evolution algorithm is that a coarse GP model is used for fitness calculation and selection, and at most two solutions will be evaluated using the real objective function at each generation, which are chosen based on a second surrogate, a fine RBF model. Note that an archive is used to store all solutions that are evaluated using the real fitness function. In the following, we will give a detailed description of the training of the coarse GP model and the fine RBF model, GP-assisted environmental selection, RBF-assisted individual refinement, and RBF-assisted local optimization.

Construction of the coarse Gaussian process

As the Gaussian process model is used for the selection of parents for the next generation, the archive samples distributed in the neighborhood of the current parent and the trial/offspring populations are to be used for constructing the coarse GP model. To minimize the impact of unevenly distributed training samples that result in poor approximation quality of the GP model in certain regions of the whole design space, a training sample selection strategy based on the maximin distance criterion with the current global best to be the initially selected sample is employed to determine the final training set. Here it should be noted that the coarse GP model aims to capture the global fitness landscape related to the location of the current parent and the trial/offspring populations. The procedure to train a coarse GP model is given in Algorithm 2.

In Algorithm 2, $pop(t)$ and $pop_{\text{trial}}(t)$ are the parent population and the trial/offspring population, respectively. $MP(t)$ represents the population merged by $pop(t)$ and $pop_{\text{trial}}(t)$. CNs denotes a set that includes all neighbors of the solutions in the merged population $MP(t)$. Generally, a minimum of $D + 1$ training samples are required to train the GP model

Algorithm 1 Pseudo code of the MADE algorithm

- 1: Offline solution sampling: generate a number of solutions using SLHD, evaluate their fitness using the expensive real objective function, and save them in the archive;
- 2: Population initialization: choose NP best solutions from the archive to form the initial population $pop(t)$ of DE, set $t = 0$;
- 3: **while** the computational budget is not exhausted **do**
- 4: Generate a trial population: generate a trial population $pop_{\text{trial}}(t)$ by means of mutation and crossover from the current parent population $pop(t)$;
- 5: Merge the parent population $pop(t)$ and the trial (offspring) population $pop_{\text{trial}}(t)$ to obtain $MP(t) = pop(t) \cup pop_{\text{trial}}(t)$;
- 6: **Construct the Gaussian process model**; (see Algorithm 2)
- 7: **Conduct the GP-assisted environmental selection** (see Algorithm 3), and determine the parent population $pop(t + 1)$;
- 8: **Train the radial basis function model**; (see Algorithm 4)
- 9: **Implement the RBF-assisted individual refinement**: calculate the fitness of all individuals in population $pop(t + 1)$ using the RBF model, and screen the ones superior to the global best for real-evaluation;
- 10: **if** the current global best is unimproved **then**
- 11: **Perform the RBF-assisted local optimization**: search the optimum of the RBF model in the local crossover region covered by the parent population $pop(t + 1)$ and the best evaluated samples;
- 12: Calculate the fitness of the optimum using the real fitness function;
- 13: **end if**
- 14: Update the current global best with the newly evaluated solutions;
- 15: Update the archive by adding the newly evaluated solutions in the archive, and set $t = t + 1$;
- 16: **end while**

Algorithm 2 Construction of the coarse GP model

- 1: **for** each solution in the merged population $MP(t)$, either in $pop(t)$ or $pop_{\text{trial}}(t)$, **do**
- 2: Find $2 \times (D + 1)$ closest neighboring solutions (nb_i) in the archive;
- 3: **end for**
- 4: Store the closest neighbors of all solutions in the $MP(t)$ to CNs , and eliminate the duplicated solutions;
- 5: Set the training set $T_s = \phi$;
- 6: Find the solution in CNs that has the best fitness value, and save it to set T_s ;
- 7: **while** $|T_s| \leq 2 \times (D + 1)$ **do**
- 8: **for** each solution in $CNs - T_s$ **do**
- 9: Find the closest neighbor in set T_s , and denote the distance to the closest neighbor as $dist_k$, $k = 1, 2, \dots, |CNs - T_s|$
- 10: **end for**
- 11: Put the k -th solution in set $CNs - T_s$ that satisfies $\max_{k=1,2,\dots,|CNs-T_s|} dist_k$ into T_s ;
- 12: **end while**
- 13: Construct GP model using dataset T_s .

for a D -dimensional optimization problem [51] so that the resulting model is able to properly learn the global profile of the fitness landscape. As suggested in [3,16], the number of training samples should not be less than $2 \times D$, so we set the size of training set to $2 \times (D + 1)$ in our work.

In Algorithm 2, we obtain the final training samples with two steps. First, a number of closest neighbors are found for

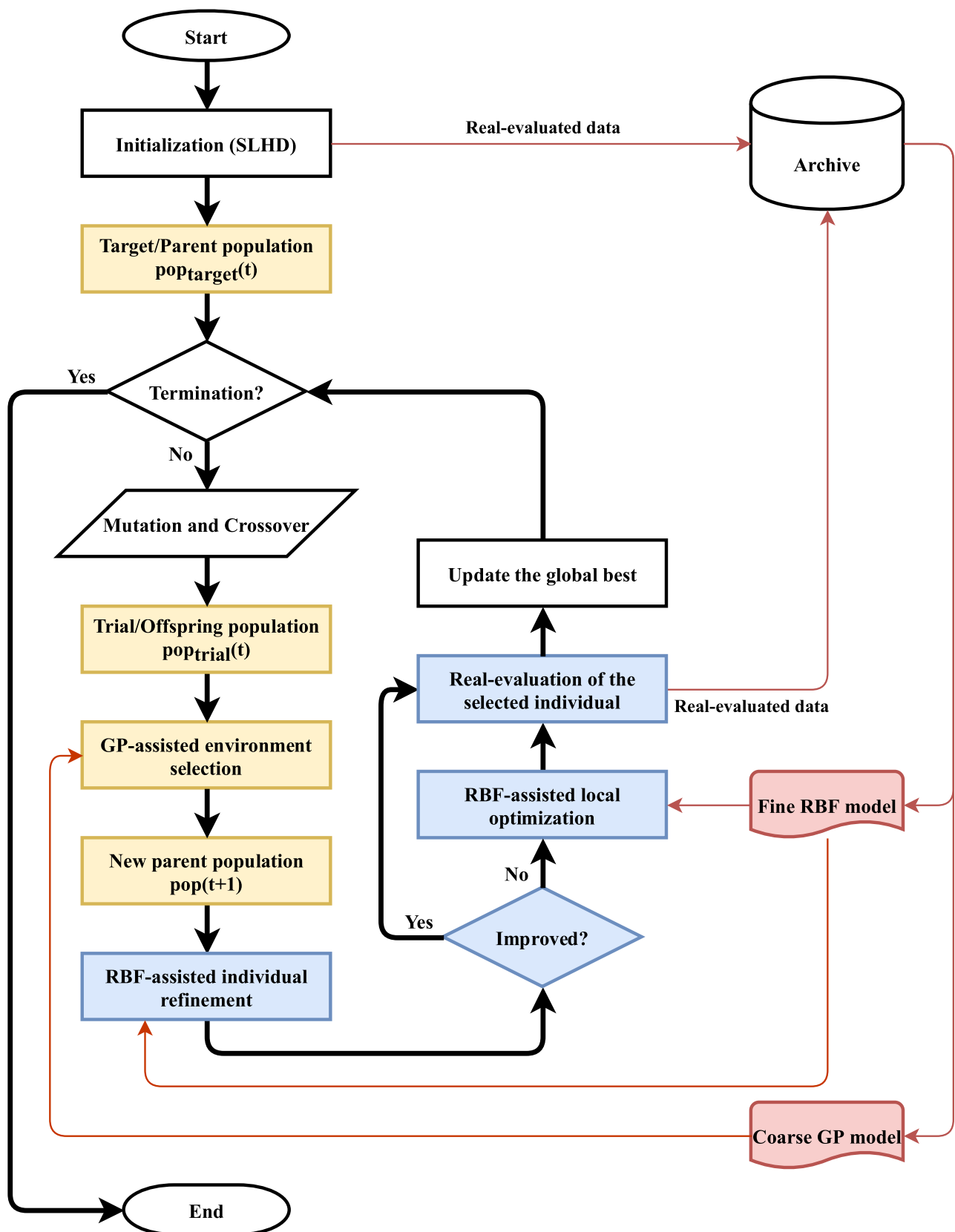


Fig. 1 A diagram of the multi-surrogate assisted differential evolution algorithm

each solution in the merged population. Then, $2 \times (D + 1)$ samples will be sequentially selected according to the distance to the best solution in the neighborhood. Figure 2 gives an illustrative example showing the procedure of sample selection using a two-dimensional Ackley function. In this work, $2 \times (2 + 1) = 6$ samples are required for training the coarse GP model. In this example, to better show the proposed strategy for selecting training samples, 12 samples will be chosen. In the figure, the stars in black represent all individuals in the merged population, including the parent and trial/offspring populations. The circles in red are a union of all neighbors found for each individual in the merged population, and the triangles in blue are the selected samples for training the GP. The numbers in Fig. 2 represent the order of individuals in the merged neighbors chosen to be saved in the training set T_s for training the GP model. The solution indicated with the number 1 has the best fitness value among all merged neighbors, so it will be saved in set T_s at first, denoted as $T_s = \{\mathbf{x}_1\}$. And then the minimum distance of each individual in the merged neighbors to the current set T_s will be calculated. We can see that solution number 2 has the maximum value among all these minimum distances, which will be saved to T_s , $T_s = \{\mathbf{x}_1, \mathbf{x}_2\}$. We continue to do this for the rest individuals in the merged neighbors and this time, solution number 3 has the maximum distance to T_s and will be saved to set $T_s = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$. This procedure continues until this is done for all solutions in the merged neighbors. From Fig. 2, we can see that the training samples will be widely distributed in the decision subspace where the merged population is located. We can also note that different markers overlap at some positions in Fig. 2. This is because the individual in the parent population may be evaluated using the real objective function, which will be saved in the archive, and obviously will be one of the individuals in the merged neighbors because of the zero-distance to the merged population.

GP-assisted environmental selection

The main benefit of using the GP model is that it can provide not only the approximated fitness value but also an estimated uncertainty of the approximated fitness. Different from most existing DE algorithms that use the fitness value to select parents for the next generation, the proposed MADE algorithm selects parents based on both the approximated fitness and the estimated uncertainty. More specifically, solutions with a better fitness value will be passed to the next generation. Besides, we also expect that some individuals with large estimated uncertainty can be included in the next population to avoid searching for potentially poor solutions and achieve a good trade-off between exploration and exploitation. Accordingly, bi-objective selection will be performed as described in Eq. (1):

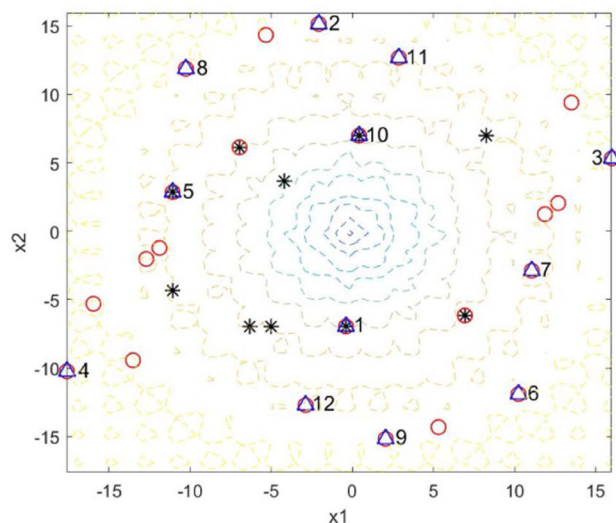


Fig. 2 An illustrative example of data selection for constructing the coarse GP model

$$\min \mathbf{g}(\mathbf{x}) = (\hat{f}(\mathbf{x}), -\hat{s}(\mathbf{x})) \quad (1)$$

In Eq. (1), $\hat{f}(\mathbf{x})$ and $\hat{s}(\mathbf{x})$ are the approximated fitness and estimated uncertainty, respectively. Algorithm 3 lists the pseudo-code of selecting parents for the next generation. The merged population will be sorted into a number of non-dominated fronts at first according to the two objectives, i.e., the approximated fitness and estimated uncertainty, basing on the non-dominated sorting in [12]. Then we select parent solutions from the first non-dominated front until the next parent population is filled. If the number of solutions on the last front to be selected is more than what the population can hold, only the ones with larger crowding distances [12] will be selected.

Algorithm 3 GP-assisted environmental selection

- 1: Sort the individuals that are in the merged population $MP(t)$ using the fast non-dominated sorting approach proposed in [12];
 - 2: Set $pop = \phi$. Assume there are NF fronts in total and each front has FS_i , $i = 1, 2, \dots, NF$ solutions;
 - 3: **for** $i = 1 : NF$ **do**
 - 4: **if** $|pop| + |FS_i| < NP$ **then**
 - 5: $pop = pop \cup FS_i$;
 - 6: **else**
 - 7: Sort the solutions on front i based on the crowding distance values in an descending order;
 - 8: Select the first $NP - |pop|$ solutions and combine them with pop ;
 - 9: **end if**
 - 10: **end for**
 - 11: Output pop .
-

Fine RBF-assisted individual refinement

After the parents for the next generation are selected, a fine surrogate model, an RBF model, is built to re-approximate the fitness of these parents so that the approximation quality of the parent individuals can be refined. Herein, the parents with superior RBF prediction against the current global best will undergo the real objective function evaluation. The strategy for the fine RBF modeling is described in Algorithm 4. As depicted in Algorithm 4, all the nearest neighbors surrounding the selected parent population $pop(t + 1)$ without duplicates are taken to train the fine RBF model.

Algorithm 4 Radial Basis Function Modeling

- 1: **for** each individual pop_i in the current parent population $pop(t + 1)$ **do**
 - 2: Find its closest $2 \times (D + 1)$ neighbors from the archive;
 - 3: **end for**
 - 4: Store the neighbors of all solutions in $pop(t + 1)$ to CN_{SRBF} , eliminate the duplicate solutions;
 - 5: Train the RBF model using dataset CN_{SRBF} .
-

Fine RBF-assisted local optimization

When there is no parent solution that is superior to the current global best after refining the approximation quality of the selected parent population according to the fine RBF model, an optimization based on this fine RBF model is performed in the local regions that are intersections of the hypercube areas covered by the partial top-ranking real-evaluated samples and the selected parent population. For the fine RBF-assisted local optimization, a multi-point based swarm intelligence optimizer SL-PSO [7] is used as the underlying local search engine instead of the traditional single-point based gradient-based optimization algorithm, such as sequential quadratic optimization (SQP) [46], to ensure sufficient exploitation of the local fitness landscape, avoiding premature convergence due to the initial value sensitivity. However, other swarm intelligence optimizers can also be used for the fine RBF-assisted local optimization.

Some practical issues

In MADE, the coarse GP-assisted environmental selection, the fine RBF-assisted individual refinement and the fine RBF-assisted local optimization are consecutively used to balance the trade-off between global exploration and local exploitation. Here attention should be paid to the roles of different surrogate models, where the coarse GP model is employed to provide additional guidance information on the uncertainty of the trial/offspring population for the identification of the new parent population in the environmental

selection instead of the one-to-one competitive selection operator in the canonical DE, whereas the fine RBF model is employed to identify better individuals amongst the selected parents and to find the optimal solutions in the neighborhood covered by the selected parents and the best-evaluated samples. For the GP-assisted environmental selection, promising parent individuals for the consecutive evolution are sifted out from the merged population of the parent population coupled with the trial/offspring population, according to their approximated fitness and approximated uncertainty. These individuals driven by the DE behavior learning operators (DE/current-to-best/1 used in this paper) are prone to gradually explore the unknown landscape of the whole search space, especially the ones with large approximated uncertainty. On the other side, for the RBF-assisted local optimization, the optimum of the RBF model is used to update the current global best and accelerate the convergence of the algorithm, thereby facilitating the local exploitation. And meanwhile, these optima can also be provided as additional training sample alternatives for GP modeling as well as for improving the quality of the RBF model in the optimal region.

Additionally, it is conceivable that more and more sample points will aggregate in the optimal region of the design space as the optimization proceeds since the sequential infilling of the real-evaluated optimal solutions determined in the RBF-assisted individual refinement and the RBF-assisted local optimization in each generation, especially the ones in the fine RBF-assisted local optimization. Thus, in this paper, a distance threshold

$$\varepsilon = \min \left(\sqrt{10^{-6}D}, 5.0 \times 10^{-5}D \times \min(UB - LB) \right) \quad (2)$$

in [37] is preset as an additional selection criterion in determining whether the candidate solutions will be re-evaluated using the real objective function so as to avoid the evaluated solutions being too close to each other, where D is the problem dimension, UB and LB denote the upper and lower bounds of the search space, respectively. Here in the RBF-assisted individual refinement and RBF-assisted local optimization, the optimal solutions whose distances to the real-evaluated samples are larger than the threshold will undergo re-evaluation by the real objective function.

Experimental results and discussions

To evaluate the effectiveness and scalability of MADE, two sets of benchmark problems, including five commonly used basic test functions [58,65] featured by symmetric fitness landscape with optimum lying on the origin and eight benchmark problems from CEC's 14 expensive optimization test suit [38] characterized by non-symmetric fitness landscape and shifted optimum (non-origin optimum), are adopted in

Table 1 Characteristics of the benchmark problems

Problem	Dimension	Characters	Global optimum ($f(\mathbf{x}^*)$)
Ellipsoid	10, 20, 30	Unimodal	0.0
Rosenbrock	10, 20, 30	Multimodal with narrow valley	0.0
Ackley	10, 20, 30	Multimodal	0.0
Griewank	10, 20, 30	Multimodal	0.0
Rastrigin	10, 20, 30	Multimodal	0.0
Shifted Sphere Function (F_1 – F_3 in [38])	10, 20, 30	Unimodal	0.0
Shifted Ellipsoid Function (F_4 – F_6 in [38])	10, 20, 30	Unimodal	0.0
Shifted and Rotated Ellipsoid Function (F_7 – F_9 in [38])	10, 20, 30	Unimodal	0.0
Shifted Step Function (F_{10} – F_{12} in [38])	10, 20, 30	Unimodal	0.0
Shifted Ackley's Function (F_{13} – F_{15} in [38])	10, 20, 30	Multimodal	0.0
Shifted Griewank's Function (F_{16} – F_{18} in [38])	10, 20, 30	Multimodal	0.0
Shifted Rotated Rosenbrock's Function (F_{19} – F_{21} in [38])	10, 20, 30	Multimodal with narrow valley	0.0
Shifted Rotated Rastrigin's Function (F_{22} – F_{24} in [38])	10, 20, 30	Multimodal	0.0

the comparative studies. Table 1 lists the 13 test functions and their main characteristics. We conduct experiments on these test problems of 10, 20 and 30 dimensions, respectively, therefore, a total of 39 benchmark problems are involved.

In the following subsections, we first examine the effectiveness of the GP and RBF models in MADE, followed by a verification of the validity of the local search. Then we perform an investigation on the behavioral characteristics of MADE and finally make a comparison of its performance against several state-of-the-art algorithms. In the experiments, the population size of all compared DE algorithms are set to $5 \times D$ for problems of 10, 20, and 30 dimensions, and the stopping criterion is when the maximum number of fitness evaluation reaches $11 \times D$. The scaling factor F and crossover rate Cr are set to $F = 0.50$ and $Cr = 0.75$, respectively [4]. The SL-PSO used in the fine RBF-assisted local optimization follows the same parameter settings as suggested in the corresponding literature, and the termination condition of SL-PSO is when the maximum number of iteration reaches $50 \times D$ or the local optimum of the fine RBF model remains unchanged for 20 consecutive generations. 30 independent runs are carried out for each test instance and all compared algorithms are implemented in MATLAB® R2019b and run on an Intel(R) Core(TM) i7-9700U CPU @3.00 GHz desktop.

The roles of GP model and RBF model

In this work, we use a GP model for selecting parents for the next generation and an RBF model for searching the best solutions to be evaluated using the real objective function. To show the effectiveness of the GP model and the RBF model, here we compare the proposed MADE algorithm that using both GP and RBF models with two MADE variants, i.e., MADE-GPOnly that using GP model alone in MADE and MADE-RBFOOnly that using RBF model alone in MADE, on the five basic benchmark problems. The characteristics of these two algorithms and MADE are summarized in Table 2. Here it should be noted that the main differences between these three algorithms lie in three aspects: surrogate-assisted environmental selection, surrogate-assisted individual refinement and surrogate-assisted local optimization. Other settings of these three algorithms such as initialization, training sample selection, and the update of the global best, etc. follow Algorithm 1.

Table 3 lists the statistical results of these three algorithms on the five basic benchmark problems over 30 independent runs. In Table 3, symbols “+”, “–”, “≈” indicate that MADE-GPOnly and MADE-RBFOOnly perform significantly worse than, significantly better than and comparably with MADE, respectively, according to the pairwise Wilcoxon rank-sum test at 0.05 significance level. The “Win/Loss/Tie” records the number of benchmark instances that MADE wins,

Table 2 Characteristics of MADE, MADE-GPOnly and MADE-RBFOnly

Algorithms	Environmental selection	Individual refinement	Local optimization
MADE	<ul style="list-style-type: none"> Using a coarse GP model Using non-dominated sorting to determine the new parent population from the merged population 	<ul style="list-style-type: none"> Using a fine RBF model Selecting the individual with the best RBF prediction among the new parent population for real-evaluation 	<ul style="list-style-type: none"> Using the same fine RBF model as in the individual refinement Local search based on SL-PSO
MADE-GPOnly	<ul style="list-style-type: none"> Using a coarse GP model Using non-dominated sorting to determine the new parent population from the merged population 	<ul style="list-style-type: none"> Using a fine GP model Selecting the individual with the best GP prediction among the new parent population for real-evaluation 	<ul style="list-style-type: none"> Using the same fine GP model as in the individual refinement Local search based on SL-PSO
MADE-RBFOnly	<ul style="list-style-type: none"> Using a coarse RBF model Using the top-ranking individuals among the merged population to constitute the new parent population 	<ul style="list-style-type: none"> Using a fine RBF model Selecting the individual with the best RBF prediction among the new parent population for real-evaluation 	<ul style="list-style-type: none"> Using the same fine RBF model as in the individual refinement Local search based on SL-PSO

losses and ties with MADE-GPOnly and MADE-RBFOnly. In general, MADE can obtain better and comparable results against the other two contestants on the selected benchmark problems. Comparing to MADE-GPOnly, MADE can obtain significantly better results on all selected benchmark problems throughout 10, 20 and 30 dimensions, indicating the superiority of the fine RBF model against the fine GP model in improving the performance of MADE. Comparing to MADE-RBFOnly, MADE wins on six out of 15 benchmark problems and tied on nine problems. Specifically, MADE can obtain better results on a majority of benchmark problems except for the Griewank function, on which the results obtained by MADE-RBFOnly are slightly better than those of MADE. We speculate that it may be due to the smooth effect of the cubic kernel function used in the fine RBF model that smoothed out the local optima, thus facilitating the fast convergence of the parent population. However, from Table 3, we can find that MADE can obtain significantly better results than MADE-RBFOnly on Rosenbrock function and Ackley function throughout 10, 20 and 30 dimensions, indicating the effectiveness and good performance of the coarse GP-assisted environmental selection in MADE.

As a whole, the results in Table 3 confirm the effectiveness and efficiency of the coarse GP and the fine RBF model in MADE, and also demonstrated the superiority of the coarse-to-fine evaluation scheme based on these two surrogate models.

The effectiveness of local search

In MADE, we perform a local optimization to search the optimum of the fine RBF model for real-evaluation when the current global best is unimproved. To verify the effectiveness of local search, we count the number of local searches performed by MADE on the five basic benchmark problems for 30 runs. Figure 3 gives the boxplot of the number of executions of local search by MADE on 10-, 20- and 30-dimensional cases. In general, from Fig. 3, we can find that the number of local searches is within half of the total computational budget of 11D real fitness function evaluations for 10- to 30-dimensional problems. The number of executions of local search increases with the number of dimensions. In addition, from Fig. 3, we can also note that the deviations of the number of local searches from 30 runs are large on the Ellipsoid function and Griewank function of 10–30 dimensions. This can be attributed to the single funnel landscape features of these two problems. Overall, the results in Fig. 3 verified the effectiveness of local search.

Behavioral analysis of MADE

In this subsection, we compare the proposed MADE with several MADE variants concerning RBF-DE, RBF-DE-LS,

Table 3 Statistical results of MADE, MADE-GPOnly and MADE-RBFOnly on the selected benchmark problems

Problem	D	MADE-RBFOnly	MADE-GPOnly	MADE
Ellipsoid	10	1.84E−02 ± 6.79E−02(≈)	2.51E+00 ± 4.75E+00(+)	1.20E−02 ± 5.07E−02
Ellipsoid	20	8.36E−03 ± 1.55E−02(≈)	5.46E+01 ± 7.65E+01(+)	2.34E−03 ± 7.83E−03
Ellipsoid	30	5.01E−03 ± 1.25E−02(≈)	9.57E+01 ± 1.12E+02(+)	1.78E−03 ± 3.88E−03
Rosenbrock	10	1.58E+01 ± 1.30E+01(+)	5.59E+01 ± 3.24E+01(+)	1.17E+01 ± 9.90E+00
Rosenbrock	20	3.68E+01 ± 2.09E+01(+)	1.36E+02 ± 5.70E+01(+)	2.47E+01 ± 1.26E+01
Rosenbrock	30	5.62E+01 ± 3.15E+01(+)	2.21E+02 ± 9.97E+01(+)	4.14E+01 ± 2.17E+01
Ackley	10	5.39E+00 ± 4.67E+00(+)	1.58E+01 ± 6.07E+00(+)	3.62E+00 ± 4.40E+00
Ackley	20	5.06E+00 ± 4.56E+00(+)	1.90E+01 ± 7.91E−01(+)	3.30E+00 ± 3.37E+00
Ackley	30	4.53E+00 ± 3.77E+00(+)	1.92E+01 ± 4.01E−01(+)	2.01E+00 ± 1.52E+00
Griewank	10	4.83E−01 ± 3.21E−01(≈)	3.12E+00 ± 4.01E+00(+)	6.88E−01 ± 3.68E−01
Griewank	20	1.24E−01 ± 2.03E−01(≈)	8.45E+00 ± 1.36E+01(+)	3.52E−01 ± 2.99E−01
Griewank	30	7.77E−02 ± 1.18E−01(≈)	2.40E+01 ± 3.33E+01(+)	1.71E−01 ± 2.17E−01
Rastrigin	10	3.19E+01 ± 2.36E+01(≈)	4.72E+01 ± 2.30E+01(+)	2.80E+01 ± 2.43E+01
Rastrigin	20	5.36E+01 ± 4.33E+01(≈)	1.23E+02 ± 2.38E+01(+)	4.47E+01 ± 3.94E+01
Rastrigin	30	5.93E+01 ± 5.46E+01(≈)	2.07E+02 ± 3.95E+01(+)	6.54E+01 ± 6.59E+01
Win/Lose/Tie		6/0/9	15/0/0	–

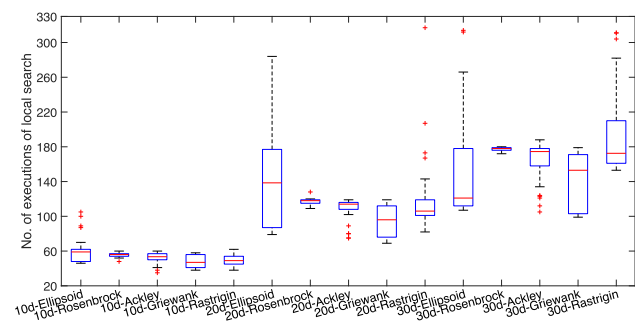


Fig. 3 The number of local searches performed by MADE on the selected benchmark problems for 30 runs

GP_{EI}-DE, GP_{MinF}-DE, MADE-NoLS, and the standard DE, to gain further insight into the impact of each algorithm component on the performance of MADE, such as the coarse GP-assisted environmental selection, fine RBF-assisted individual refinement, and RBF-assisted local optimization.

Description of MADE variants

Table 4 shows the main characteristics of MADE variants in the environmental selection and infill-sampling. More specifically, for the RBF-DE, this method is an RBF-assisted DE, in which an RBF model is built to predict the fitness of the derived trial/offspring population after performing mutation and crossover operations on the parent population. The training samples for RBF modeling are drawn from the adjacent evaluated samples surrounding the trial/offspring population. In the model management, the infilling strategy follows a greedy pairwise comparison sampling rule that identifies the

individuals in the trial/offspring population with superior predictions over their associated parents as candidate solutions for real-evaluation. And these selected individuals will be real-evaluated only when their distances to the evaluated samples meet the predefined distance criterion, which is the same as the one adopted in MADE. Note that, in the environmental selection of RBF-DE, the new parent population for the next generation is determined through a pairwise competition rule that selects better individuals from each pair of target/parent and trial/offspring individuals.

For the RBF-DE-LS, it is an RBF-DE variant that embedded an RBF-assisted SL-PSO local search after the environmental selection. The RBF model adopted in the local search is the same as the one in the environmental selection. As with the MADE, the region for local search is also the intersection area of the hypercube neighborhoods surrounded by the trial/offspring population and the best sample subset in the archive. And the optimum found in local search performs real-evaluation only if its distance from the evaluated samples meets the predefined distance criterion.

For the GP_{EI}-DE and GP_{MinF}-DE, these two algorithms are two variants of MADE stripped of the RBF-assisted SL-PSO local search, both of them employ only a GP model for assisting the environmental selection and infill sampling. GP_{EI}-DE and GP_{MinF}-DE adopt the same GP modeling method that collects the nearest neighbors surrounding the parent and trial/offspring populations as training samples for constructing the GP model. And both algorithms use the same environmental selection strategy proposed in the MADE. The key point to differentiating between these two algorithms lies in the infill criterion, where GP_{MinF}-DE relies

Table 4 Characteristics of MADE variants

Algorithms	Environmental selection	Infill-sampling
RBF-DE	<ul style="list-style-type: none"> Using a fine RBF model trained by the adjacent evaluated samples of the trial/offspring population Using a pairwise competition rule to select better individuals from each pair of parent and offspring individuals to determine the new parent population 	<ul style="list-style-type: none"> Selecting the individuals in the trial/offspring population with superior predictions over their associated parents for real-evaluation
RBF-DE-LS	<ul style="list-style-type: none"> Using a fine RBF model trained by the adjacent evaluated samples of the trial/offspring population Using a pairwise competition rule to select better individuals from each pair of parent and offspring individuals to determine the new parent population 	<ul style="list-style-type: none"> Selecting the individuals in the trial/offspring population with superior predictions over their associated parents for real-evaluation Searching the optimum of the fine RBF model by SL-PSO for real-evaluation
GP _{MinF} -DE	<ul style="list-style-type: none"> Using a coarse GP model Using non-dominated sorting to determine the new parent population from the merged population 	<ul style="list-style-type: none"> Selecting the individual with the minimum GP prediction for real-evaluation
GP _{EI} -DE	<ul style="list-style-type: none"> Using a coarse GP model Using non-dominated sorting to determine the new parent population from the merged population 	<ul style="list-style-type: none"> Selecting the individual maximizing the expected improvement criterion for real-evaluation
MADE-NoLS	<ul style="list-style-type: none"> Using a coarse GP model Using non-dominated sorting to determine the new parent population from the merged population 	<ul style="list-style-type: none"> Using a fine RBF model Selecting the individual with the best RBF prediction among the new parent population for real-evaluation

on the performance-based sampling that chooses the individual with the minimum GP prediction in the parent population for real-evaluation, while GP_{EI}-DE chooses the individual maximizing the EI criterion.

For MADE-NoLS, it is also an MADE variant that excludes the RBF-assisted local optimization part in MADE. As with the MADE, MADE-NoLS adopts a GP-assisted environmental selection operator to determine the parent population for the next generation and builds a fine RBF model to filter out the individual with the best predictions among the new parent population for subsequent real-evaluation.

Comparison results of MADE and MADE variants

The statistical results averaged over 30 independent runs are listed in Table 5, wherein the best value on each benchmark instance is highlighted in boldface. The pairwise Wilcoxon rank-sum test [17, 18] with 95% confidence level is conducted for investigation of significant differences between MADE

and MADE variants, and symbols “+”, “−”, “≈” indicate that MADE performs significantly better than, significantly worse than, and comparably with the compared MADE variants, respectively. The average rank of each algorithm is calculated according to the Friedman test [13], and the “Win/Loss/Tie” records the number of benchmark instances that MADE wins, losses, and ties with the contenders.

From Table 5, one can observe that all surrogate-assisted DE algorithms can obtain better results than the canonical DE within a limited computational budget. Compare MADE with RBF-DE, MADE obtains the best solutions on all benchmark problems varying from different dimensionalities and fitness landscapes, indicating the positive effects of associating multiple surrogates and local search to improve the performance of MADE. Comparing MADE with RBF-DE-LS, MADE obtains the best results on the majority of test instances except for the Griewank function, on which the RBF-DE-LS performs the best. This may be attributed to the low precision of the coarse GP model that fails to capture the local details of the multi-peak topology on the fitness

Table 5 Comparison results of MADE and MADE variants on the basic benchmark problems of 10, 20, and 30 dimensions

Problem	D	DE	RBF-DE	RBF-DE-LS	GP _{MinF} -DE	GP _{EP} -DE	MADE-NoLS	MADE
Ellipsoid	10	8.77E+01 ± 3.08E+01(+)	8.53E+01 ± 2.59E+01(±)	1.92E+01(±)	1.36E+01(±)	3.26E+01 ± 3.26E+01(±)	1.20E-02 ± 5.07E-02	
Ellipsoid	20	5.21E+02 ± 8.92E+01(±)	1.06E+02(±)	6.88E+01(±)	7.26E+01(±)	1.05E+02(±)	2.34E-03 ± 7.83E-03	
Ellipsoid	30	1.35E+03 ± 1.85E+02(±)	2.04E+02(±)	8.98E+01 ± 8.98E+01(±)	2.02E+02(±)	3.18E+02(±)	1.78E-03 ± 3.88E-03	
Rosenbrock	10	3.82E+02 ± 1.56E+02(±)	1.27E+02(±)	1.28E+02(±)	7.41E+01(±)	1.09E+02(±)	1.17E+01 ± 9.90E+00	
Rosenbrock	20	1.39E+03 ± 3.77E+02(±)	3.99E+02(±)	4.60E+02(±)	1.39E+02(±)	2.61E+02(±)	2.47E+01 ± 1.26E+01	
Rosenbrock	30	3.03E+03 ± 6.64E+02(±)	6.62E+02(±)	4.61E+02(±)	4.25E+02(±)	4.87E+02(±)	4.14E+01 ± 2.17E+01	
Ackley	10	1.76E+01 ± 1.23E+00(±)	2.40E+00(±)	5.0E+01 ± 5.70E+00(±)	4.22E+00(±)	5.3E+01 ± 4.54E+00(±)	3.62E+00 ± 4.40E+00	
Ackley	20	1.90E+01 ± 1.01E+00(±)	4.92E-01(±)	6.68E+00(±)	1.92E+00(±)	8.8E+01 ± 1.40E+00(±)	3.30E+00 ± 3.37E+00	
Ackley	30	1.96E+01 ± 5.11E-01(±)	9.4E+01 ± 5.14E-01(±)	2.6E+01 ± 8.37E+00(±)	9.7E+01 ± 1.27E+00(±)	9.7E+01 ± 4.48E-01(±)	2.01E+00 ± 1.52E+00	
Griewank	10	6.83E+01 ± 2.04E+01(±)	4.0E+01 ± 2.22E+01(±)	4.6E-07 ± 9.67E-07(±)	0.9E+01 ± 1.06E+01(±)	0.4E+01 ± 1.69E+01(±)	6.88E-01 ± 3.68E-01	
Griewank	20	1.84E+02 ± 3.43E+01(±)	7.7E+02 ± 3.30E+01(±)	9.8E-06 ± 4.67E-06(±)	2.1E+01 ± 2.62E+01(±)	5.9E+01 ± 4.23E+01(±)	3.52E-01 ± 2.99E-01	
Griewank	30	3.25E+02 ± 4.64E+01(±)	9.7E+02 ± 5.65E+01(±)	7.9E-06 ± 1.01E-06(±)	0.7E+02 ± 5.33E+01(±)	5.3E+02 ± 5.75E+01(±)	1.71E-01 ± 2.17E-01	
Rastrigin	10	9.47E+01 ± 1.36E+01(±)	4.1E+01 ± 9.85E+00(±)	9.8E+01 ± 2.04E+01(±)	0.3E+01 ± 1.80E+01(±)	8.5E+01 ± 2.28E+01(±)	2.80E+01 ± 2.43E+01	
Rastrigin	20	2.19E+02 ± 1.73E+01(±)	1.7E+02 ± 1.67E+01(±)	9.1E+02 ± 7.23E+01(±)	0.67E+02 ± 3.81E+01(±)	8.5E+02 ± 3.20E+01(±)	4.47E+01 ± 3.94E+01	
Rastrigin	30	3.45E+02 ± 2.66E+01(±)	3.5E+02 ± 4.72E+01(±)	8.1E+02 ± 1.20E+02(±)	5.2E+02 ± 4.93E+01(±)	7.9E+02 ± 4.65E+01(±)	6.54E+01 ± 6.59E+01	
Average rank	6.9	6.0	2.9	3.7	4.7	2.6	1.2	
Win/Lose/Tie	15/0/0	15/0/0	12/0/3	15/0/0	15/0/0	15/0/0	15/0/0	–

landscape of the Griewank function covered by the merged population, resulting in the failed selection of the correct offspring population. However, the results demonstrate the attractive performance of GP-assisted environmental selection in guiding the right convergence of MADE. Comparing with another two MADE variants, i.e., $GP_{\text{MinF-DE}}$ and $GP_{\text{EI-DE}}$, we can find from Table 5 that the MADE-NoLS have achieved significantly better results on most of the benchmark instances, revealing the good performance of fine RBF-assisted individual refinement in enhancing the right prescreening of promising individuals for real-evaluation. When comparing to MADE, however, MADE-NoLS obtains significantly worse results on all the benchmark instances, verifying the excellent reinforcement effect of RBF-assisted local optimization in substantially improving the performance of MADE. In addition, from Table 5, in terms of the average ranking according to the Friedman test, for all involved benchmark functions, MADE gets the first ranking, followed by MADE-NoLS, RBF-DE-LS, $GP_{\text{MinF-DE}}$, $GP_{\text{EI-DE}}$, RBF-DE, and canonical DE. As a conclusion, the results in Table 5 demonstrated the good synergy between the components of MADE in steering its correct convergence.

The convergence profiles of the compared algorithm on the selected benchmark problems of 10, 20, and 30 dimensions are summarized in Fig. 4, which were averaged over 30 independent runs. From Fig. 4, we can find that MADE performs the best on Ellipsoid, Rosenbrock, Ackley, and Rastrigin function through 10–30 dimensions, whereas the RBF-DE-LS outperforms other alternatives on Griewank function. For the Griewank function characterized by a multi-peaked single funnel topology, MADE adopts a coarse GP model to select a promising parent population for guiding the evolving dynamics of DE at each iteration, which is conducive to smoothing the search space and accelerating convergence in the early stage of optimization as shown in Fig. 4j–l. Nonetheless, the coarse GP model fails to capture the multi-peak fitness landscape around the global optimum, resulting in the incorrect selection of the parent population and premature convergence in the later search stage. In contrast, RBF-DE-LS constructs a fine RBF model to learn a priori knowledge concerning the local features of the global optimal region, thus delivering high accuracy in the selection of a new parent population. Note that in MADE the fine RBF model is utilized to refine the new parent population and sift out the best-predicted individual for real-evaluation during the RBF-assisted individual refinement, while in RBF-DE-LS, the fine RBF model is performed to screen the predicted trail/offspring individuals that are better than the associated target/parent individuals so as to identify the parent population for the next generation.

In terms of $GP_{\text{MinF-DE}}$ and $GP_{\text{EI-DE}}$, $GP_{\text{MinF-DE}}$ outperforms $GP_{\text{EI-DE}}$ on most of the benchmark instances, especially for Rosenbrock and Rastrigin function. $GP_{\text{EI-DE}}$

performs slightly better than $GP_{\text{MinF-DE}}$ on the 10-dimensional Ackley function, while both present comparable performance on the 20- and 30-dimensional Ackley function. From Fig. 4, we can also find that both $GP_{\text{MinF-DE}}$ and $GP_{\text{EI-DE}}$ perform better than RBF-DE on all the selected benchmark instances through 10–30 dimensions and poses a good convergence tendency, demonstrating the effectiveness of GP-assisted environmental selection operator in the parent population selection.

From Fig. 4, however, one can note that among the three MADE variants without RBF-assisted local optimization, MADE-NoLS outperforms $GP_{\text{MinF-DE}}$ and $GP_{\text{EI-DE}}$ on all benchmark instances, corroborating the imperative of using the fine RBF model to refine the approximation quality of the selected parent population. Here note that the correct filtration of the parent population contributes to providing more valuable training samples for the subsequent update of the GP model.

For Rosenbrock and Rastrigin function, from Fig. 4d–f and 4m–o, we can find that RBF-DE-LS are outperformed by MADE-NoLS throughout 10- to 30-dimensional instances. And RBF-DE-LS stagnates and prematurely converges in the later stage of optimization. We surmise that the excessive consumption of real-evaluations causes the performance degradation of RBF-DE-LS since RBF-DE-LS consumes a majority of real-evaluations for selecting the new parent population, whereas no real-evaluation is conducted in MADE-NoLS during the GP-assisted environmental selection and at most one individual performs real-evaluation in RBF-assisted individual refinement at each iteration.

From Fig. 4, we can also find that MADE-NoLS achieves a dramatic performance improvement in conjunction with the fine RBF-assisted local optimization (i.e., MADE) on all the benchmark instances, and MADE maintains a good downtrend against other alternatives, indicating the effectiveness of the RBF-assisted local optimization in promoting the exploitation capability of MADE. To summarize, the results shown above demonstrate the effectiveness and feasibility of collaboration between coarse GP-assisted environmental selection, fine RBF-assisted individual refinement, and RBF-assisted local optimization in MADE.

Performance comparison between isomorphic algorithms

To examine the efficacy of MADE with other isomorphic algorithms, in this subsection, we conduct two sets of comparative studies of MADE with four representative state-of-the-art algorithms, i.e., CAL-SAPSO [65], GORS-SSLPSO [70], FSAPSO [37], and S-JADE [4], on two different types of benchmark test suits, respectively. Here the CEC'14 expensive optimization test suit featured by non-symmetric fitness landscapes with shifted global optima is

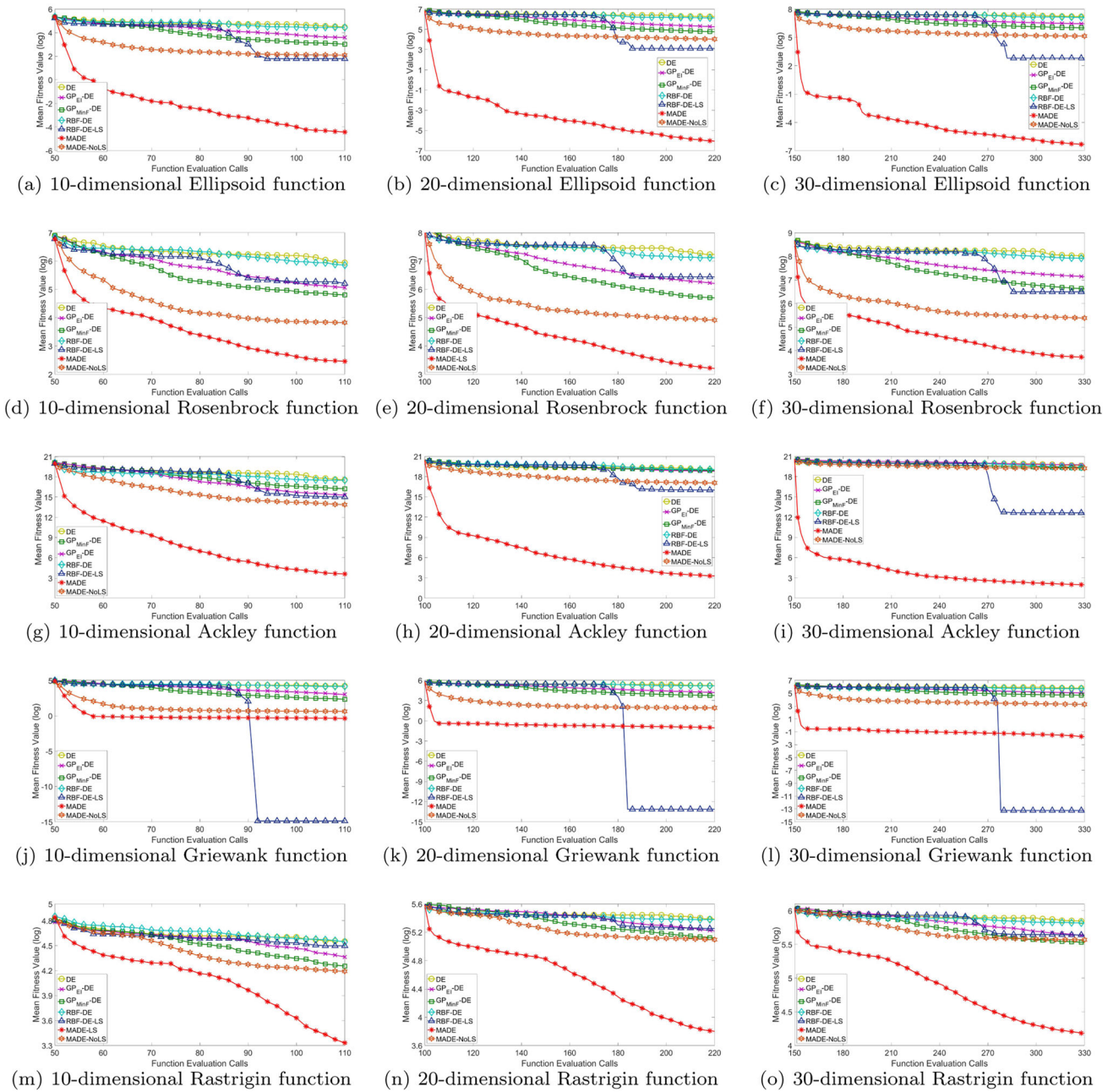


Fig. 4 Convergence profiles of compared algorithms on the selected benchmark problems of 10, 20, and 30 dimensions

used as another test suit apart from the above five basic test functions. The main task of these comparisons is to gain insight into the performance differences of these methods in handling different types of problems within a small number of real fitness function evaluations.

In the following subsections, we first compare MADE with CAL-SAPSO, GORS-SSLPSO, and FSAPSO on the five popular basic test functions, followed by a comparison of MADE with GORS-SSLPSO, FSAPSO, and S-JADE on the CEC'14 expensive optimization test suit. To make a fair comparison, these four algorithms adopt the same popula-

tion size of 5D and each of them terminates after exhausting the computational budget of 11D real fitness function evaluations, thus ensuring that each algorithm consumes the same number of real-evaluations. The other parameters involved are configured with the same settings as recommended in their corresponding literature.

Illustrations on the algorithm architectures of these five contestants

CAL-SAPSO [65] sequentially employed one multi-surrogate ensemble to search for the most uncertain solution and the other for finding the best-approximated solution for real-evaluation in the whole design space, while adopting a local multi-surrogate ensemble to exploit the optimal region covered by the top-ranking archive samples for seeking the promising optima to be real-evaluated depending on whether the optimal solution achieves improvement. Analogous to CAL-SAPSO, MADE takes full advantage of multiple surrogates and leverages the uncertainty information provided by the coarse GP model for sifting out promising parent population for the next generation, whilst refining the best-approximated individual for real-evaluation by a fine RBF model. An RBF-assisted local search is also integrated within MADE to exploit the promising region surrounding by the top-ranking archive samples. The key to distinguishing CAL-SAPSO from MADE lies in the fact that the former concentrates on sampling promising solutions for re-evaluation by the multi-surrogate ensemble-based PSO search in the global exploration part, while the latter puts emphasis on selecting a promising parent population by means of a coarse-to-fine model evaluation for guiding the global exploration of DE.

GORS-SSLPSO [70] built a global RBF model to approximate the design space and employed the SL-PSO [7] to locate the optimal solution, where the search population of SL-PSO was periodically adjusted for consecutively approaching and exploiting the optimal region. GORS-SSLPSO dynamically restarted the underlying SL-PSO with a top-ranking archive sample subset to be the new initial population, and the optimum found in each restart loop was selected for real-evaluation. As distinct from GORS-SSLPSO, MADE determines the parent population for the next generation through a non-dominated sorting in the light of the uncertainty information of candidate solutions derived from a coarse GP model. On the other side, as with the MADE, GORS-SSLPSO also performs the local search centered around the neighborhood of the top-ranking archive sample subset in each restart loop. Therefore, it is interesting to compare the performance of these two approaches to gain certain insight into the efficacy of different environmental selection strategies.

FSAPSO [37] searched the optimum of the global RBF model in the territory covered by the iterative swarm and the best predicted particle amongst the iterative swarm for real-evaluations in tandem in each generation. And in case the global optimum remains unimproved, the most uncertain particle amongst the iterative swarm was additionally selected as a candidate solution for real-evaluation, wherein the uncertainties of particles were calculated by weighting the distance

and fitness of their adjacent evaluated samples. FSAPSO has been proved to perform well on small and medium-scale problems. Similar to MADE, FSAPSO also incorporates an RBF-assisted local optimization, where a global RBF model was constructed using all the evaluated samples and the search region was circumscribed to a local region covered by the current iterative swarm. MADE differs from FSAPSO mainly in that MADE draws on the uncertainty information provided by the coarse GP model in environmental selection to extract promising parent population for the next generation and performs local optimization resorting to a fine local RBF model and a multi-point swarm intelligence optimizer SL-PSO, emphasizing generating instructive parent population, whereas FSAPSO focuses on infill sampling from the candidate solutions.

S-JADE [4] combined global and local RBF models to guide the mutation and selection operations in the optimization process. The optima of the global and local RBF models were separately utilized as competitive alternatives to the randomly chosen top-ranking best and dissimilar population individuals in the mutation operator for guiding the mutation operation. And in the environmental selection, a portion of trial/offspring individuals with smaller predictions by the global RBF model amongst the trial/offspring population underwent real-evaluation and further competed with the associated parent individuals for entering the next generation, while the remaining parent individuals were directly inherited from the ones in the previous generation. It follows that S-JADE employs the one-to-one rivalry selection to update part of the parent population, whereas MADE updates the parent population by a coarse GP-assisted environmental selection. And unlike S-JADE that constructs a local RBF model for each individual and uses the optima of both global and local RBF models to guide the mutation direction, MADE only uses the optimum of a local RBF model as an alternative to the current global best to guide the mutation of the target/parent population.

Comparison results of algorithms on the five basic test problems

Table 6 lists the optimization results of the four algorithms under comparison on the five basic test problems, including the mean and variance (shown as Mean \pm Std.) obtained by each algorithm after 30 independent runs, the pairwise Wilcoxon rank-sum test at a significant level of 0.05, and the average rank of each algorithm calculated by Friedman test, where the symbols “+”, “−” and “ \approx ” indicate that MADE performs significantly superior to, significantly inferior to, and comparable to the contestants. As seen from Table 6, MADE can obtain significantly better results than CAL-SAPSO and FSAPSO throughout 10- to 30-dimensional benchmark problems within a limited evaluation

budget. Comparing to GORS-SSLPSO, MADE acquires significantly better results on Ellipsoid, Rosenbrock, and Ackley functions. For Griewank and Rastrigin function, from Table 6, we can see that there is no significant difference in the results of GORS-SSLPSO and MADE except for the 30-dimensional Griewank function, on which MADE gets a significantly better result. We speculate that this may be due to the smoothing effect of the coarse GP model in MADE, inducing the parent population to quickly converge to the global optimal region, therefore providing a larger probability to locate the global optimum. Furthermore, according to the average rank, MADE gets the first rank, followed by GORS-SSLPSO, FSAPSO, and CAL-SAPSO. Thus, it can be concluded that for the basic test functions characterized by symmetric landscapes and optima lied on the origin, MADE can be chosen as a promising optimizer in contrast to GORS-SSLPSO, FSAPSO, and CAL-SAPSO when only a small number of real fitness evaluations are available, e.g., 11D real-evaluation budget.

Figure 5 presents the convergence profiles of these four approaches on the selected benchmark functions. As seen from Fig. 5a–c and 5g–i, MADE outperforms the other three contestants remarkably on the unimodal Ellipsoid and multimodal Ackley functions of 10–30 dimensions under an 11D real-evaluation budget. For the Ellipsoid function as shown in Fig. 5a–c, GORS-SSLPSO performs comparably to FSAPSO, whereas GORS-SSLPSO outperforms FSAPSO on the multimodal Ackley, Griewank, and Rastrigin functions. Nonetheless, both of them are outperformed by MADE as shown in Fig. 5g–o.

For the Rosenbrock function featured by multimodal and narrow valley fitness landscape, MADE converges slowly in the early stage of optimization, where CAL-SAPSO and GORS-SSLPSO converges faster as shown in Fig. 5d–f. It is conceivable that in the early search stage the surrogate ensemble in CAL-SAPSO approximates the fitness landscape with high precision under limited training samples available and the top-ranking solutions in GORS-SSLPSO provide a promising leading population. However, MADE and FSAPSO exhibit good downward trends in the later search stage, where MADE outperforms the other three compared algorithms on 10- and 20-dimensional cases and performs comparably with FSAPSO on 30-dimension case as shown in Fig. 5d–f.

For the single-funnel Griewank function characterized by symmetric and regular multimodal fitness landscape, MADE converges slightly faster than the other three compared approaches, especially on the 30-dimensional case as shown in Fig. 5j–l, yet there is no significant difference between MADE and GORS-SSLPSO on 10- and 20-dimensional cases according to Table 6. Likewise, for the multimodal Rastrigin function with multiple funnels, MADE performs comparably to GORS-SSLPSO and both

perform superior to CAL-SAPSO and FSAPSO across 10- to 30-dimensional cases within 11D real fitness function evaluations. These results confirm the fact that the selection of suitable evolutionary populations is conducive to improving the convergence performance of the algorithm while enhancing its adaptability to different fitness landscapes.

Comparison results of algorithms on the CEC'14 expensive optimization test suit

Table 7 lists the comparison results of MADE, S-JADE, FSAPSO and GORS-SSLPSO on the CEC'14 expensive optimization test suit within 11D real fitness function evaluations. In general, MADE significantly outperforms S-JADE, FSAPSO and GORS-SSLPSO on 21, 16 and 11 instances out of 24 benchmark problems, respectively, and MADE gets the first rank, followed by GORS-SSLPSO, FSAPSO and S-JADE, according to the Friedman test.

For the two surrogate-assisted DE variants MADE and S-JADE, the results in Table 7 show that MADE can obtain significantly better results than S-JADE on nine instances (F1–F9) among 12 unimodal benchmark problems (F1–F12), on which MADE is comparable to S-JADE on F10 and F11, and is outperformed by S-JADE on F12. Although there is no significant difference between the results obtained by MADE and S-JADE on F10 and F11, the mean values of MADE averaged over 30 independent runs are worse than those of S-JADE. However, from Table 7, MADE significantly outperforms S-JADE on 12 multimodal benchmark problems (F13–F24), indicating the superiority of MADE against S-JADE in dealing with problems with non-symmetric fitness landscapes and shifted global optima under a small number of real fitness function evaluations.

In terms of the two surrogate-assisted PSO variants FSAPSO and GORS-SSLPSO, from Table 7, we can find that MADE has obtained significantly better results than GORS-SSLPSO on six out of 12 unimodal benchmark problems (i.e., F3, F6, F10, F11 and F12) and five out of 12 multimodal benchmark problems (i.e., F14, F15, F19, F20, and F21), especially on the shifted Step function (F10–F12) and shifted rotated Rosenbrock's function across 10–30 dimensionalities. And on the other side, MADE ties with GORS-SSLPSO on 11 out of 24 benchmark problems, and is inferior to GORS-SSLPSO on 20-D F17 and 10-D F22. Comparing MADE to FSAPSO, from Table 7, one can also find that MADE has obtained significantly better results than FSAPSO on all the 12 multimodal benchmark problems, while the results obtained by MADE are as good as those of FSAPSO on six out of 12 unimodal instances and are worse than those of FSAPSO on shifted sphere function of 20 and 30 dimensions (F2–F3). As a whole, the results show in Table 7 further demonstrate the remarkable performance of MADE against GORS-SSLPSO and FSAPSO on problems with different fitness landscapes

Table 6 Statistics of CAL-SAPSO, GORS-SSLPSO, FSAPSO, and MADE on the basic test functions of 10–30 dimensions

Problem	D	CAL-SAPSO	GORS-SSLPSO	FSAPSO	MADE
Ellipsoid	10	1.14E+00 ± 9.95E-01 (+)	5.31E-02 ± 5.35E-02 (+)	3.84E-02 ± 3.87E-02 (+)	1.20E-02 ± 5.07E-02
Ellipsoid	20	1.55E+00 ± 5.15E-01 (+)	2.32E-01 ± 2.12E-01 (+)	1.98E-01 ± 1.80E-01 (+)	2.34E-03 ± 7.83E-03
Ellipsoid	30	3.29E+00 ± 1.47E+00 (+)	6.11E-01 ± 3.85E-01 (+)	7.17E-01 ± 5.72E-01 (+)	1.78E-03 ± 3.88E-03
Rosenbrock	10	1.59E+01 ± 3.54E+00 (+)	1.77E+01 ± 8.08E+00 (+)	1.44E+01 ± 5.75E+00 (+)	1.17E+01 ± 9.90E+00
Rosenbrock	20	3.53E+01 ± 5.57E+00 (+)	5.45E+01 ± 1.54E+01 (+)	3.73E+01 ± 1.38E+01 (+)	2.47E+01 ± 1.26E+01
Rosenbrock	30	4.94E+01 ± 9.98E+00 (+)	9.67E+01 ± 2.14E+01 (+)	4.34E+01 ± 1.45E+01 (+)	4.14E+01 ± 2.17E+01
Ackley	10	1.99E+01 ± 4.43E-01 (+)	4.32E+00 ± 2.77E+00 (+)	6.64E+00 ± 3.15E+00 (+)	3.62E+00 ± 4.40E+00
Ackley	20	2.01E+01 ± 3.61E-15 (+)	4.77E+00 ± 1.92E+00 (+)	7.67E+00 ± 3.96E+00 (+)	3.30E+00 ± 3.37E+00
Ackley	30	1.60E+01 ± 5.62E-01 (+)	4.98E+00 ± 1.28E+00 (+)	7.24E+00 ± 2.19E+00 (+)	2.01E+00 ± 1.52E+00
Griewank	10	1.13E+00 ± 1.14E-01 (+)	8.02E-01 ± 1.58E-01 (≈)	9.08E-01 ± 1.08E-01 (+)	6.88E-01 ± 3.68E-01
Griewank	20	1.05E+00 ± 3.57E-02 (+)	3.85E-01 ± 1.35E-01 (≈)	7.59E-01 ± 1.25E-01 (+)	3.52E-01 ± 2.99E-01
Griewank	30	9.92E-01 ± 4.36E-02 (+)	3.65E-01 ± 1.13E-01 (+)	6.62E-01 ± 1.14E-01 (+)	1.71E-01 ± 2.17E-01
Rastrigin	10	8.55E+01 ± 2.37E+01 (+)	2.86E+01 ± 1.93E+01 (≈)	4.37E+01 ± 2.17E+01 (+)	2.80E+01 ± 2.43E+01
Rastrigin	20	7.35E+01 ± 1.53E+01 (+)	4.66E+01 ± 2.01E+01 (≈)	6.94E+01 ± 3.20E+01 (+)	4.47E+01 ± 3.94E+01
Rastrigin	30	9.08E+01 ± 1.93E+01 (+)	7.00E+01 ± 3.06E+01 (≈)	8.25E+01 ± 3.62E+01 (+)	6.54E+01 ± 6.59E+01
Average rank		3.73	2.53	2.73	1
Win/Lose/Tie		15/0/0	10/0/5	15/0/0	–

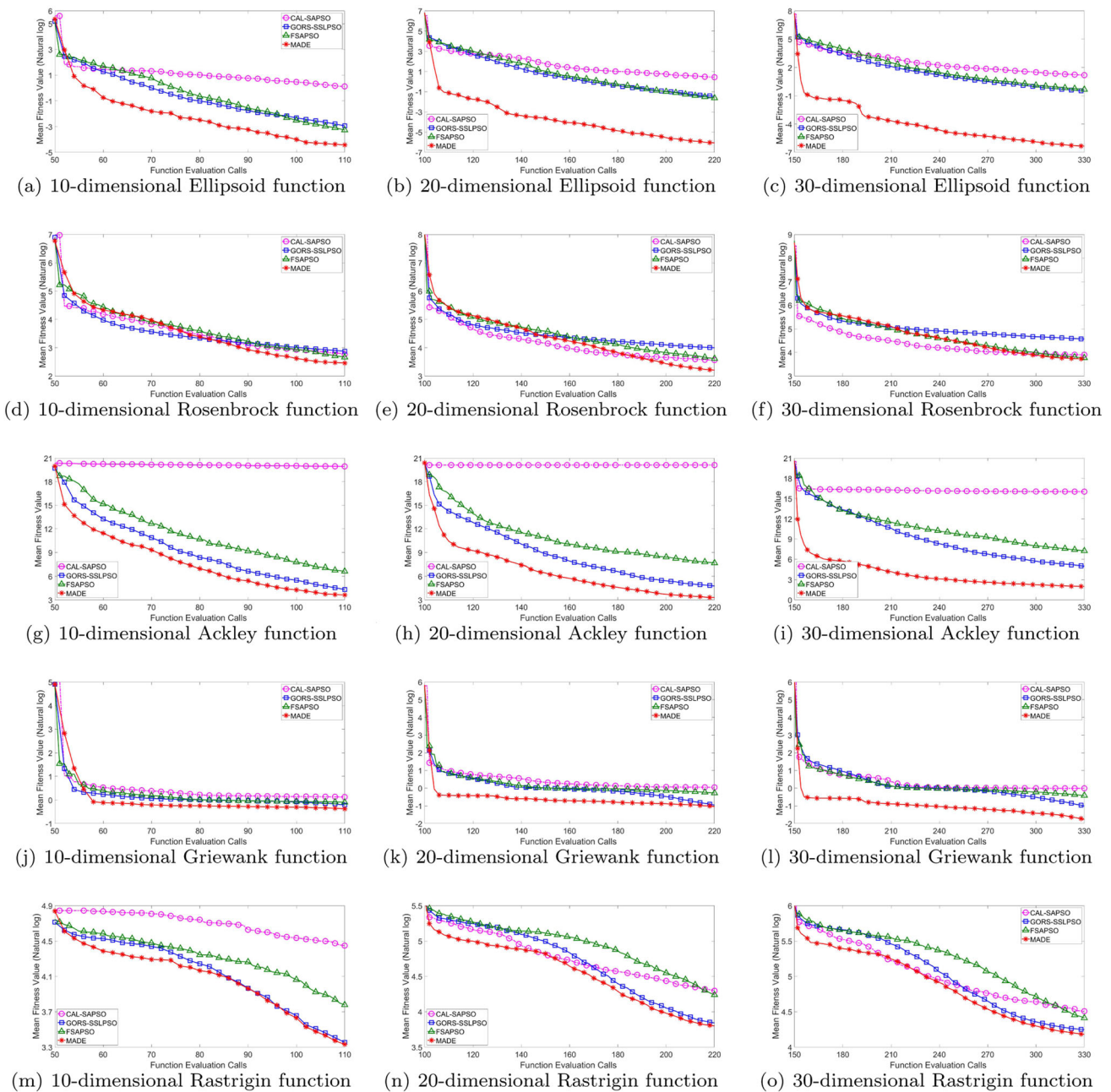


Fig. 5 Convergence profiles of compared algorithms on the basic test problems of 10, 20, and 30 dimensions

and non-origin optimum when only a limited real-evaluation budget is available.

Figures 6, 7 provides the convergence profiles of these four contestants on the CEC'14 expensive optimization test suit. From Fig. 6, it can be observed that S-JADE is significantly outperformed by the other three contenders on four unimodal benchmark problems of 10 to 30 dimensions including the shifted sphere function (F1–F3), shifted Ellipsoid function (F4–F6), and shifted and rotated Ellipsoid function (F7–F9). This may be due to the fact that S-JADE consumes more real-evaluations in the environmental selec-

tion than the other three contenders in each generation. In contrast, the performance of GORS-SSLPSO degrades on the shifted Step function of 10, 20 and 30 dimensions and GORS-SSLPSO converges prematurely in 11D real-evaluations, whereas S-JADE performs slightly better than MADE and FSAPSO, and MADE outperforms FSAPSO with the increase of dimensionality. We surmise that GORS-SSLPSO stagnates on the discontinuous and flat basins owing to the greedy search of population composed of the optimal subset, resulting in the failure of escaping from the local traps, while S-JADE invokes more real-evaluations to eval-

Table 7 Statistics of GORS-SSLPSO, FSAPSO, S-JADE and MADE on the CEC'14 expensive optimization test suit

Problem	GORS-SSLPSO	FSAPSO	S-JADE	MADE
F1	3.33E−05 ± 4.81E−05 (≈)	5.53E−05 ± 7.17E−05(≈)	3.53E−01 ± 2.10E−01(+)	1.32E−04 ± 2.29E−04
F2	2.55E−05 ± 3.08E−05(≈)	1.06E−05 ± 1.66E−05 (−)	1.80E+00 ± 7.62E−01(+)	5.84E−05 ± 7.10E−05
F3	6.09E−05 ± 5.85E−05(+)	5.29E−06 ± 7.31E−06 (−)	7.31E+00 ± 2.05E+00(+)	5.76E−05 ± 1.32E−04
F4	3.15E+00 ± 3.47E+00(≈)	1.94E+00 ± 1.60E+00 (≈)	1.83E+01 ± 1.25E+01(+)	2.33E+00 ± 2.76E+00
F5	1.69E+01 ± 1.18E+01(≈)	1.82E+01 ± 1.10E+01(≈)	1.48E+02 ± 6.89E+01(+)	1.68E+01 ± 8.80E+00
F6	4.32E+01 ± 1.47E+01(+)	4.18E+01 ± 1.69E+01(+)	4.53E+02 ± 1.21E+02(+)	3.10E+01 ± 1.59E+01
F7	4.13E+01 ± 3.15E+01(≈)	3.21E+01 ± 3.05E+01(≈)	1.66E+02 ± 1.02E+02(+)	2.96E+01 ± 2.56E+01
F8	8.36E+01 ± 3.80E+02(+)	1.54E+01 ± 1.13E+01(+)	1.78E+02 ± 7.56E+01(+)	1.22E+01 ± 1.71E+01
F9	2.01E+02 ± 1.18E+02(≈)	2.34E+02 ± 1.26E+02(+)	1.67E+03 ± 5.40E+02(+)	1.62E+02 ± 1.10E+02
F10	5.47E+00 ± 5.24E+00(+)	2.10E+00 ± 2.16E+00 (≈)	1.87E+00 ± 1.22E+00 (≈)	2.77E+00 ± 1.94E+00
F11	1.57E+01 ± 1.14E+01(+)	9.07E+00 ± 5.30E+00(≈)	5.60E+00 ± 2.04E+00 (≈)	6.67E+00 ± 4.23E+00
F12	4.22E+01 ± 3.15E+01(+)	2.02E+01 ± 1.28E+01(+)	1.29E+01 ± 3.95E+00 (−)	1.58E+01 ± 6.26E+01
F13	3.99E+00 ± 2.84E+00(≈)	6.71E+00 ± 2.88E+00(+)	9.02E+00 ± 1.50E+00(+)	2.97E+00 ± 1.10E+00
F14	5.15E+00 ± 3.14E+00(+)	7.81E+00 ± 2.67E+00(+)	9.23E+00 ± 9.10E−01(+)	3.79E+00 ± 2.70E+00
F15	6.81E+00 ± 2.70E+00(+)	9.03E+00 ± 3.05E+00(+)	9.24E+00 ± 7.85E−01(+)	3.79E+00 ± 2.45E+00
F16	8.01E−01 ± 2.19E−01(≈)	9.06E−01 ± 1.29E−01(+)	1.05E+00 ± 5.68E−02(+)	7.61E−01 ± 1.95E−01
F17	4.10E−01 ± 1.43E−01 (−)	8.08E−01 ± 1.21E−01(+)	1.21E+00 ± 7.77E−02(+)	5.77E−01 ± 1.87E−01
F18	3.39E−01 ± 1.11E−01 (≈)	6.54E−01 ± 1.07E−01(+)	2.03E+00 ± 2.65E−01(+)	3.40E−01 ± 2.19E−01
F19	1.57E+02 ± 7.78E+01(+)	3.38E+02 ± 1.50E+02(+)	3.01E+02 ± 1.41E+02(+)	7.09E+01 ± 3.71E+01
F20	1.30E+02 ± 4.06E+01(+)	2.28E+02 ± 1.29E+02(+)	2.44E+02 ± 5.11E+01(+)	4.18E+01 ± 2.17E+01
F21	7.16E+02 ± 3.17E+02(+)	2.93E+03 ± 9.33E+02(+)	1.27E+03 ± 3.44E+02(+)	1.82E+02 ± 5.13E+01
F22	3.17E+01 ± 1.28E+01 (−)	5.84E+01 ± 1.97E+01(+)	7.63E+01 ± 1.03E+01(+)	4.19E+01 ± 1.97E+01
F23	5.83E+01 ± 2.21E+01 (≈)	8.04E+01 ± 2.99E+01(+)	1.62E+02 ± 1.77E+01(+)	6.28E+01 ± 2.34E+01
F24	8.21E+01 ± 2.61E+01(≈)	1.11E+02 ± 5.69E+01(+)	2.63E+02 ± 1.43E+01(+)	7.94E+01 ± 2.91E+01
Average rank	2.25	2.62	3.54	1.58
Win/Loss/Tie	11/2/11	16/2/6	21/1/2	—

uate promising offspring individuals and therefore enriches the a priori knowledge of the search space, helping it jump out of the local traps. From Fig. 6, we can also notice that MADE performs comparably with GORS-SSLPSO and FSAPSO on most of the unimodal benchmark problems except for F3 and F8, on which MADE and FSAPSO outperforms GORS-SSLPSO.

In terms of multimodal benchmark problems, it can be seen from Fig. 7 that MADE shows significant superiority against S-JADE, FSAPSO and GORS-SSLPSO on the shifted Ackley's function (F13–F15) and shifted rotated Rosenbrock's function (F19–F20) of 10, 20 and 30 dimensions within 11D real fitness function evaluations, while S-JADE performs comparably with FSAPSO on 30-D F15, 10-D F19 and 20-D F18. For the shifted Griewank function, MADE, GORS-SSLPSO and FSAPSO show comparable performance on the 10-dimensional instance, yet MADE and GORS-SSLPSO outperforms FSAPSO as the increase of dimensions, and FSAPSO performs slightly better than MADE in the later search stage. Notice that MADE performs consistently better than S-JADE on this benchmark prob-

lem across 10 to 30 dimensions, indicating the superiority of coarse GP-assisted environmental selection in economizing the invocation frequency of the real fitness function. Moreover, MADE and GORS-SSLPSO perform comparable on 20- and 30-dimensional shifted Rastrigin function (F23–F24), while the MADE is outperformed by GORS-SSLPSO on 10-D instance (F22).

In summary, the results shown in Table 7 and Figs. 6 and 7 demonstrate the good adaptability and scalability of MADE in different types of benchmark problems, which are featured by non-symmetric fitness landscapes and non-origin optima. And further corroborated the comparable and superior performance of MADE against four state-of-the-art algorithms when a small number of the real-evaluation budget is available.

Computational complexity analysis

The computational complexity of MADE consists of the following main parts according to Algorithm 1: fitness evaluation at the initialization phase, coarse GP modeling and

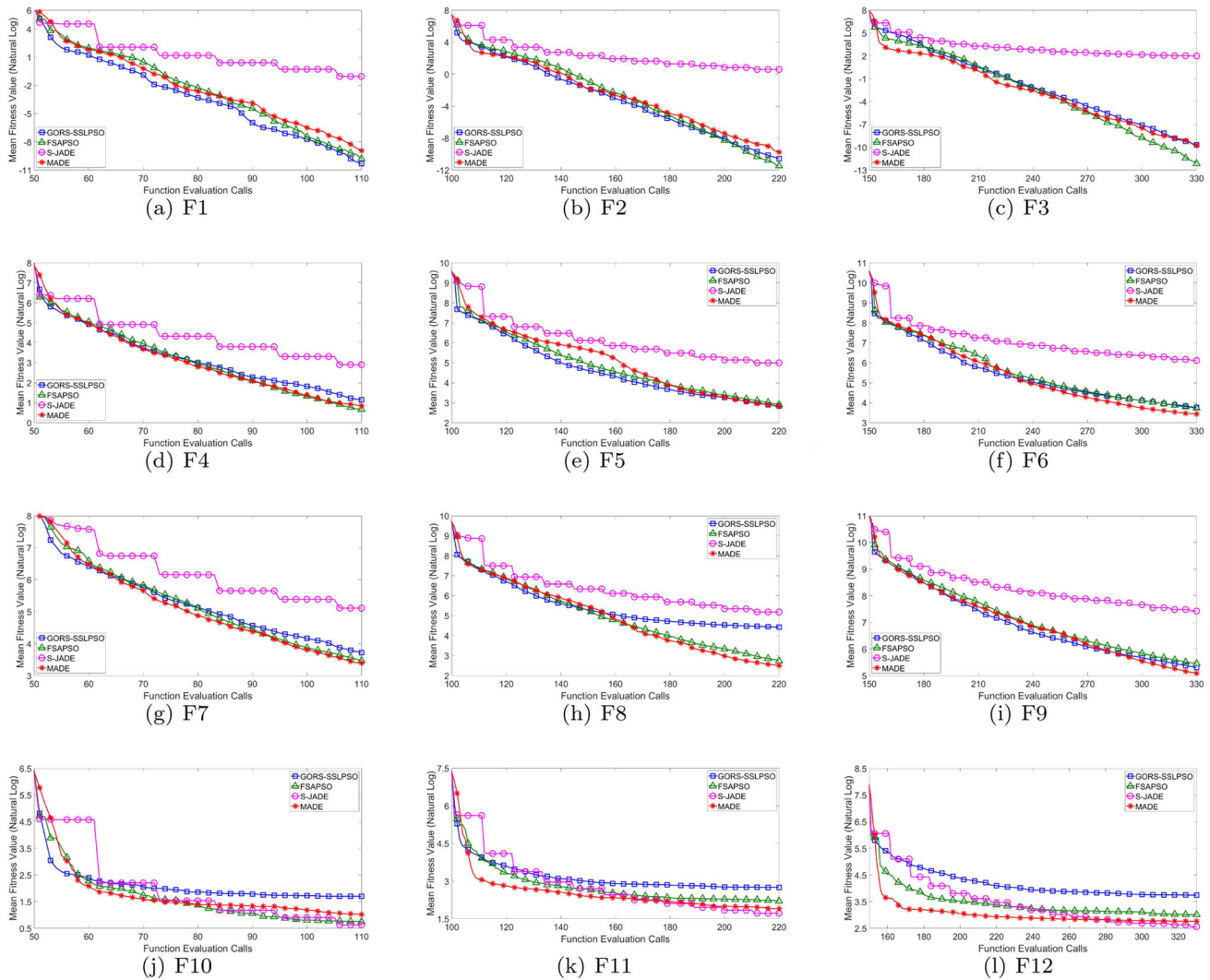


Fig. 6 Convergence profiles of GORS-SSLPSO, FSAPSO, S-JADE and MADE on the unimodal CEC'14 benchmark problems

non-dominated sorting on the merged population in GP-assisted environmental selection, fine RBF modeling and promising individual selection in RBF-assisted individual refinement, and the SL-PSO local search in RBF-assisted local optimization. In this subsection, we only give an approximation to the upper bound of the computational complexity for one generation of MADE.

At the initialization phase, the fitness evaluation of the initial population (parent population) takes $O(N_p D)$ computations, where N_p denotes the population size and D is the problem dimension. Then the parent population undergoes mutation and crossover to generate the trial/offspring population, which also requires a runtime of $O(N_p D)$. After merging the parent population and the trial/offspring population, a coarse GP model is constructed to compute the approximated fitness and estimated uncertainty of the merged population, where the training set for the coarse GP modeling is deter-

mined by a two-step sample selection. A closest sample subset of size N_{ns} to the merged population is firstly chosen from the archive, which demands $O(2N_p N_{arc} D)$ computations for finding the closest neighbors in the archive by calculating the Euclidean distances of sample pairs. Here N_{arc} denotes the size of archive. Then N_{sgp} sample points is selected from the selected N_{ns} closest sample points to be the final training samples for the coarse GP modeling, which requires $O(N_{sgp} N_{ns} D)$ computations to calculate the Euclidean distances of sample pairs and $O(N_{ns} \log N_{ns})$ computations to sort the obtained distances. The computational complexity for building a GP model is $O(K N_{sgp}^3 D)$ [41], where K denotes the number of iterations required for hyperparameter optimization. Afterwards, the non-dominated sorting is carried out on the merged population to determine the new parent population for the next generation, which require $O(M(2N_p)^2) + O(M(2N_p) \log(2N_p))$ compu-

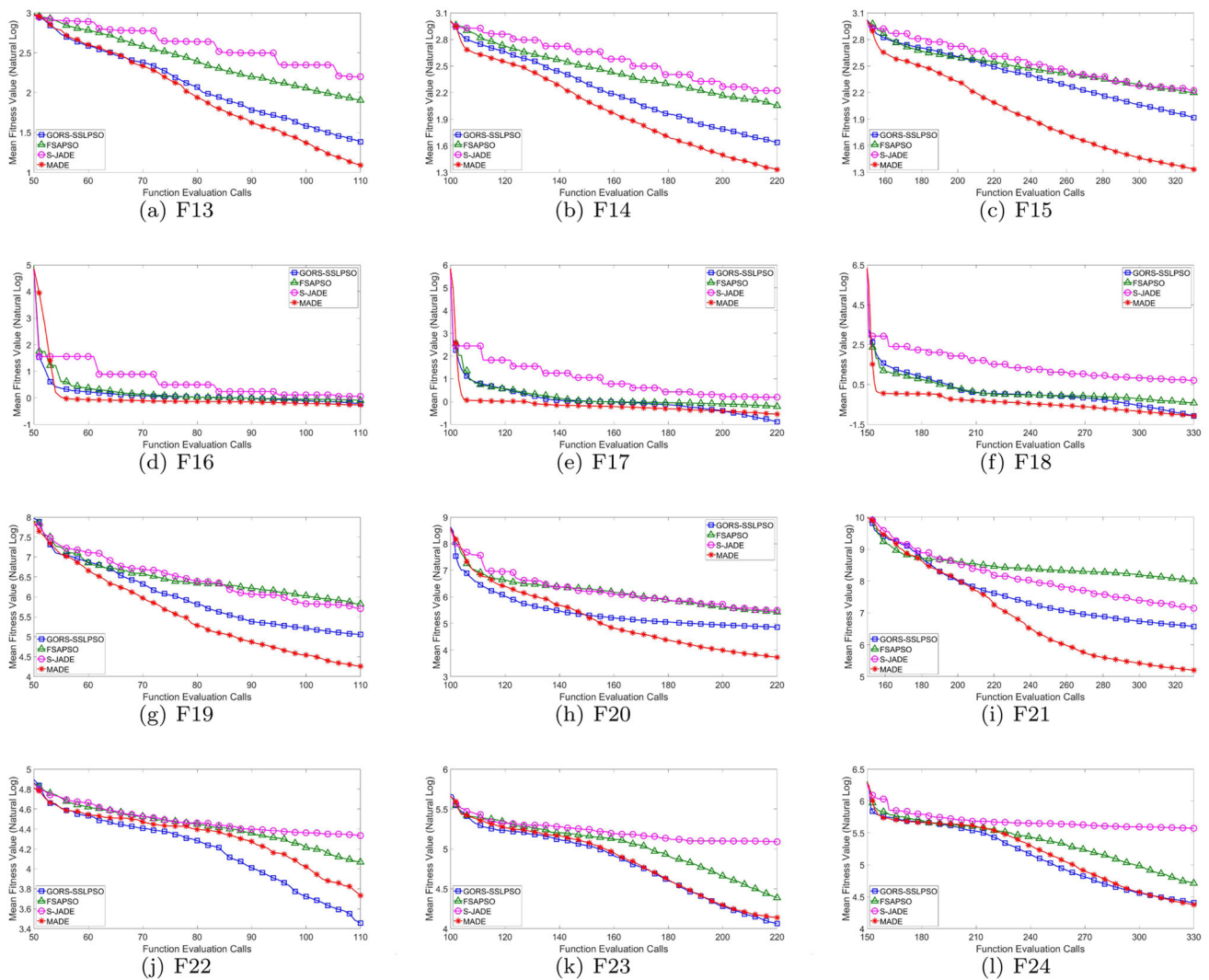


Fig. 7 Convergence profiles of GORS-SSLPSO, FSAPSO, S-JADE and MADE on the multimodal CEC'14 benchmark problems

tations. Here M denotes the number of objectives and $M = 2, M \ll N_p$ in this paper. Note that the size of the merged population is $2N_p$. Thus the total computational complexity for the GP-assisted environmental selection is $O(N_p N_{arc} D + N_{sgp} N_{ns} D + N_{ns} \log N_{ns} + K N_{sgp}^3 D)$ as N_{arc}, N_{sgp} and N_{ns} are usually bigger than N_p .

In the RBF-assisted individual refinement, a fine RBF model is built to re-approximate the selected parents from the GP-assisted environmental selection. Here a closest sample set to the new parent population is selected from the archive to be the training samples, which requires $O(N_p N_{arc} D)$ computations. To construct an RBF model of interpolation form used in our method, $O(N_{srbf}^3 D)$ computations are needed [67], where N_{srbf} denotes the size of training set. Thereafter, to select the best individual for real-evaluation, the parents are sorted according to the fitness approximated from the fine RBF model, which require

$O(N_p \log N_p)$ computations. Thus the total computational complexity for the RBF-assisted individual refinement is $O(N_p N_{arc} D + N_{srbf}^3 D + N_p \log N_p)$.

In the worst case that the current global best is unimproved, a local search based on SL-PSO is carried out for finding the optimum of the fine RBF model, which involves $O(N_{ps}^2 + N_{ps} D)$ computations [7]. Here N_{ps} denotes the population size of SL-PSO. Therefore, the overall computational complexity of MADE is as follows:

$$T = O(N_p d + N_p N_{arc} D + N_{sgp} N_{ns} D + N_{ns} \log N_{ns}) + O(K N_{sgp}^3 D + N_{srbf}^3 D + N_{ps}^2 + N_{ps} D) \quad (3)$$

Here it should be noted that the time complexity for a single fitness evaluation is non-deterministic polynomial hard (NP-hard) [47] for most of the real-world computationally

expensive problems, so the time complexity of MADE is acceptable.

Conclusion

A novel multi-model assisted differential evolution algorithm called MADE is proposed in this paper for solving computationally expensive problems. Two surrogate models, an GP model and an RBF model, is separately used to explore and exploit the design space. The GP model is constructed as a coarse surrogate to learn the global feature of the fitness landscape and assist the selection operator for selecting the instructive parent population for the next generation. An coarse GP-assisted environmental selection strategy is then developed by simultaneously considering the approximated fitness and estimated uncertainty. In addition, an RBF model was introduced to serve as a fine model to refine the approximation quality of the selected parent population while finding the best individual amongst the population and the local optimum in its vicinity for fitness evaluation using the real objective function. These two strategies together make a good balance between exploration and exploitation in searching for an optimum within a limited computation budget. The proposed algorithm was evaluated on two sets of widely used benchmark problems characterized by different types of fitness landscapes. The numerical results showed that the proposed algorithm performs better than a few state-of-the-art surrogate-assisted DE and PSO algorithms.

Model management remains the focus of research in surrogate-assisted evolutionary algorithms. As our experimental results showed that the fidelity of the surrogate model may influence the optimal results of the algorithm, in the future, the criteria for self-determining the model fidelity will be studied based on the contribution of the model in the algorithm so as to improve the performance of our method. Moreover, extending our method to multi-objective optimization and applying it for real-world computationally expensive problems are also interesting works in the future.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grant no. 61876123); Key Research and Development Program Projects in Shanxi Province (Grant no. 201703D111011), and Shanxi Province Science Foundation for Youths (Grant nos. 201901D211237 and 201901D211218), and Shanxi Science and Technology Innovation project for Excellent Talents (Grant no. 201805D211028); Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shanxi Province.

Declarations

Conflict of interest The authors declare that they have no conflict of interest that could have appeared to influence the work reported in this paper.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Differential evolution

The differential evolution [50] is used in this work as the fundamental search algorithm for the proposed framework. Generally, it is comprised of four parts, i.e., initialization, mutation, crossover, and selection. Several mutation strategies have been proposed for differential evolution [11], in this paper, the DE/current-to-best/1 mutation strategy [10], which perturbs the current target vector by adding a scaled difference of two other population members to a convex combination of the current target vector and the best population member, is adopted to generate a donor/mutant vector for each individual in the population. Therefore, a mutant vector for each individual is generated as follows:

$$\mathbf{v}_i(t) = \mathbf{x}_i(t) + F \cdot (\mathbf{x}_{\text{best}}(t) - \mathbf{x}_i(t)) + F \cdot (\mathbf{x}_{R1}(t) - \mathbf{x}_{R2}(t)) \quad (4)$$

where $\mathbf{x}_i(t)$ and $\mathbf{x}_{\text{best}}(t)$ are the i -th individual for perturbation and the best individual found so far at iteration t , respectively. $\mathbf{x}_{R1}(t)$ and $\mathbf{x}_{R2}(t)$ are two different individuals randomly chosen from the current population. $F \in (0, 2]$ denotes the scale factor which plays a role in scaling the difference vectors.

After the mutation operation for each individual, the binomial crossover operator is then conducted on the mutant vector to generate the trial vectors as follows

$$u_{ij}(t) = \begin{cases} v_{ij}(t) & \text{if } \text{rand} < Cr | j = j_{\text{rand}} \\ x_{ij}(t) & \text{otherwise} \end{cases} \quad (5)$$

where $u_{ij}(t)$ is the j -th dimension of i -th trial vector, $Cr \in (0, 1]$ denotes the crossover rate that controls the number of components inherited from the donor vector. j_{rand} is a randomly selected integer within $\{1, 2, \dots, d\}$ to ensure that at least one component of trial vector is inherited from the mutant vector.

The offspring is finally determined through a greedy selection scheme described as follows

$$\mathbf{x}_i(t+1) = \begin{cases} \mathbf{u}_i(t) & \text{if } f(\mathbf{u}_i(t)) \leq f(\mathbf{x}_i(t)) \\ \mathbf{x}_i(t) & \text{otherwise} \end{cases} \quad (6)$$

Appendix B: Gaussian process

Gaussian process (GP) [45], also called Kriging, is a type of probabilistic models, and has been developed for assisting global optimization in the past several decades [3,16,31,56]. In the Gaussian process model, it is often assumed that the responses of the training data are derived from a multivariate normal distribution. If the correlation function is defined, for example, the Gaussian correlation function which is utilized in our method, the final approximation model can be derived based on the maximum likelihood estimation and the Bayesian theory. Generally, the final approximation model consists of prediction mean $\hat{f}(\mathbf{x})$ and corresponding variance $\hat{\sigma}^2(\mathbf{x})$ which can be described as follows,

$$\hat{f}(\mathbf{x}) = \hat{\beta} + (\mathbf{y} - \mathbf{1}\hat{\beta})^T \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}) \quad (7)$$

$$\hat{\sigma}^2(\mathbf{x}) = \hat{\sigma} \left[1 - \mathbf{c}(\mathbf{x})^T \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}) + \frac{(1 - \mathbf{1}^T \mathbf{C}^{-1} \mathbf{c}(\mathbf{x}))^2}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \right] \quad (8)$$

where \mathbf{C} is the correlation matrix that each of its element is the value of the following Gaussian correlation function

$$C(\mathbf{x}_i, \mathbf{x}_j) = \exp \left[- \sum_{k=1}^d \theta_k |x_{ik} - x_{jk}|^2 \right] \quad (9)$$

The hyperparameters $\theta = (\theta_1, \theta_2, \dots, \theta_d)$, $\hat{\beta}$ and $\hat{\sigma}^2$ are estimated by maximum likelihood estimation, $\mathbf{c}(\mathbf{x}) = (C(\mathbf{x}, \mathbf{x}_1), C(\mathbf{x}, \mathbf{x}_2), \dots, C(\mathbf{x}, \mathbf{x}_N))$ is the correlation vector between the position \mathbf{x} and training dataset $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^T$, $\mathbf{1}$ is an unity vector with length D . More detailed information can be referred to [30,56].

Appendix C: Radial basis function network

The radial basis function (RBF) network, which has much faster training speed than the multilayer perceptron (MLP) trained with backpropagation (BP) rule [55], was proposed by Broomhead and Lowe [2] in 1998. Empirical findings show that the performance of the RBF network is relatively insensitive to the increase on the dimension of the problem [25]. Therefore, the radial basis function interpolation [49] is proposed to be used in this paper.

Generally, given N input–output data pairs $(\mathbf{x}_1, f(\mathbf{x}_1)), (\mathbf{x}_2, f(\mathbf{x}_2)), \dots, (\mathbf{x}_N, f(\mathbf{x}_N))$, $\mathbf{x}_i \in \mathfrak{R}^D$, $f(\mathbf{x}_i) \in \mathfrak{R}$, $i = 1, 2, \dots, N$, the radial basis function interpolates this set of points with following formula [21]:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) \quad (10)$$

where $\|\cdot\|$ and $\varphi(\cdot)$ are the Euclidian norm and basis function, respectively. Representative basis functions include cubic splines, thin-plate splines, Gaussian function, linear splines, and multi-quadrics splines [21]. In this paper, a cubic splines basis function defined as $\varphi(r) = r^3$ is used to construct the RBF model. $\alpha_i \in R$ denotes the i -th weight coefficient for cubic spline interpolation to point \mathbf{x}_i . $p(\mathbf{x})$ is a linear polynomial with D variables that satisfies $\sum_{i=1}^N \alpha_i p(\mathbf{x}_i) = 0$. The unknown parameters in Eq. (10) are then obtained by solving the following system

$$\begin{pmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{0} \end{pmatrix} \quad (11)$$

where matrix $\Phi \in \mathfrak{R}^{N \times N}$ with element $\Phi_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, $i, j = 1, 2, \dots, N$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_N)^T \in \mathfrak{R}^N$ denotes a vector of weight coefficients. $\mathbf{P} \in \mathfrak{R}^{N \times (D+1)}$ is a basis function matrix of linear polynomial $p(\mathbf{x})$ on the interpolating points, and $\mathbf{c} = (c_1, c_2, \dots, c_{D+1})^T \in \mathfrak{R}^{D+1}$ is the vector of coefficients for the linear polynomial $p(\mathbf{x})$, $\mathbf{F} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N))^T \in \mathfrak{R}^N$. Note that the coefficient matrix in Eq. (11) is nonsingular as long as the interpolating points are all affinely independent [48,52].

References

- Allmendinger R, Emmerich MTM, Hakanen J, Jin Y, Rigoni E (2017) Surrogate-assisted multicriteria optimization: complexities, prospective solutions, and business case. *J Multi-Criteria Decis Anal* 24(1–2):5–24. <https://doi.org/10.1002/mcda.1605>
- Broomhead D, Lowe D (1988) Radial basis functions, multi-variable functional interpolation and adaptive networks. Royal signals and radar establishment Malvern (United Kingdom) RSRE-MEMO-4148
- Buche D, Schraudolph NN, Koumoutsakos P (2005) Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 35(2):183–194
- Cai X, Gao L, Li X, Qiu H (2019) Surrogate-guided differential evolution algorithm for high dimensional expensive problems. *Swarm Evol Comput* 48:288–311. <https://doi.org/10.1016/j.swevo.2019.04.009>
- Cai X, Qiu H, Gao L, Shao X (2017) Metamodeling for high dimensional design problems by multi-fidelity simulations. *Struct Multidiscip Optim* 56(1):151–166
- Cai X, Qiu H, Gao L, Yang P, Shao X (2017) A multi-point sampling method based on kriging for global optimization. *Struct Multidiscip Optim* 56(1):71–88

7. Cheng R, Jin Y (2015) A social learning particle swarm optimization algorithm for scalable optimization. *Inf Sci* 291:43–60
8. Chugh T, Jin Y, Miettinen K, Hakanen J, Sindhya K (2018) A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans Evol Comput* 22(1):129–142
9. Dan G, Chai T, Jinliang D, Jin Y (2016) Small data driven evolutionary multi-objective optimization of fused magnesium furnaces. In: 2016 IEEE symposium series on computational intelligence (SSCI), pp 1–8. <https://doi.org/10.1109/SSCI.2016.7850211>
10. Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution—an updated survey. *Swarm Evol Comput* 27:1–30
11. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31. <https://doi.org/10.1109/TEVC.2010.2059031>
12. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-II. *IEEE Trans Evol Comput* 6(2):182–197
13. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
14. Eason J, Cremaschi S (2014) Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Comput Chem Eng* 68:220–232
15. Elsayed SM, Ray T, Sarker RA (2014) A surrogate-assisted differential evolution algorithm with dynamic parameters selection for solving expensive optimization problems. In: 2014 IEEE congress on evolutionary computation (CEC), pp 1062–1068. IEEE
16. Emmerich MT, Giannakoglou KC, Naujoks B (2006) Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodelling. *IEEE Trans Evol Comput* 10(4):421–439
17. Gibbons JD (1974) Nonparametric statistical methods. *Technometrics* 42(2):217–218
18. Gibbons JD, Chakraborti S (2011) Nonparametric statistical inference. *International encyclopedia of statistical science*. Springer, Berlin, pp 977–979
19. Gong W, Zhou A, Cai Z (2015) A multioperator search strategy based on cheap surrogate models for evolutionary optimization. *IEEE Trans Evol Comput* 19(5):746–758
20. Guo D, Jin Y, Ding J, Chai T (2019) Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems. *IEEE Trans Cybern* 49(3):1012–1025. <https://doi.org/10.1109/TCYB.2018.2794503>
21. Gutmann HM (2001) A radial basis function method for global optimization. *J Glob Optim* 19(3):201–227
22. Havinga J, van den Boogaard AH, Klaseboer G (2017) Sequential improvement for robust optimization using an uncertainty measure for radial basis functions. *Struct Multidiscip Optim* 55(4):1345–1363
23. Hüsken M, Jin Y, Sendhoff B (2005) Structure optimization of neural networks for evolutionary design optimization. *Soft Comput* 9(1):21–28
24. Isaacs A, Ray T, Smith W (2007) An evolutionary algorithm with spatially distributed surrogates for multiobjective optimization. In: Australian conference on artificial life. Springer, pp 257–268
25. Jin R, Chen W, Simpson TW (2001) Comparative studies of meta-modelling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13
26. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput* 9(1):3–12
27. Jin Y, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans Evol Comput* 6(5):481–494
28. Jin Y, Sendhoff B (2004) Reducing fitness evaluations using clustering techniques and neural network ensembles. *Genetic and evolutionary computation conference*. Springer, Berlin, pp 688–699
29. Jin Y, Wang H, Chugh T, Guo D, Miettinen K (2019) Data-driven evolutionary optimization: an overview and case studies. *IEEE Trans Evol Comput* 23(3):442–458. <https://doi.org/10.1109/TEVC.2018.2869001>
30. Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *J Glob Optim* 21(4):345–383
31. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13(4):455–492
32. Kattan A, Ong YS (2015) Surrogate genetic programming: a semantic aware evolutionary search. *Inf Sci* 296:345–359
33. Kenny QY, Li W, Sudjianto A (2000) Algorithmic construction of optimal symmetric latin hypercube designs. *J Stat Plan Inference* 90(1):145–159
34. Kleijnen JP (2017) Regression and kriging metamodelling with their experimental designs in simulation: a review. *Eur J Oper Res* 256(1):1–16
35. Le MN, Ong YS, Jin Y, Sendhoff B (2012) A unified framework for symbiosis of evolutionary mechanisms with application to water clusters potential model design. *IEEE Comput Intell Mag* 7(1):20–35. <https://doi.org/10.1109/MCI.2011.2176995>
36. Li E, Wang H, Ye F (2016) Two-level multi-surrogate assisted optimization method for high dimensional nonlinear problems. *Appl Soft Comput* 46:26–36
37. Li F, Shen W, Cai X, Gao L, Gary Wang G (2020) A fast surrogate-assisted particle swarm optimization algorithm for computationally expensive problems. *Appl Soft Comput* 92:106303
38. Liang J, Qu B, Suganthan P (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University Singapore*, vol 635
39. Lim D, Jin Y, Ong YS, Sendhoff B (2010) Generalizing surrogate-assisted evolutionary computation. *IEEE Trans Evol Comput* 14(3):329–355
40. Liu B, Koziel S, Zhang Q (2016) A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. *J Comput Sci* 12:28–37
41. Liu B, Zhang Q, Gielen GG (2014) A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput* 18(2):180–192
42. Liu H, Cai J, Ong YS (2017) An adaptive sampling approach for kriging metamodeling by maximizing expected prediction error. *Comput Chem Eng* 106:171–182
43. Lu XF, Tang K (2012) Classification- and regression-assisted differential evolution for computationally expensive problems. *J Comput Sci Technol* 27(5):1024–1034
44. Mallipeddi R, Lee M (2015) An evolving surrogate model-based differential evolution algorithm. *Appl Soft Comput* 34:770–787
45. Matheron G (1963) Principles of geostatistics. *Econ Geol* 58(8):1246–1266. <https://doi.org/10.2113/gsecongeo.58.8.1246>
46. Ong YS, Nair PB, Keane AJ (2003) Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA J* 41(4):687–696
47. Park J, Kim KY (2017) Meta-modeling using generalized regression neural network and particle swarm optimization. *Appl Soft Comput* 51:354–369. <https://doi.org/10.1016/j.asoc.2016.11.029>
48. Powell M (2001) Radial basis function methods for interpolation to functions of many variables. In: *HERCMA*, pp 2–24. Citeseer
49. Powell MJ (1999) Recent research at Cambridge on radial basis functions. *New developments in approximation theory*. Springer, Berlin, pp 215–232

50. Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, Berlin
51. Razavi S, Tolson BA, Burn DH (2012) Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environ Model Softw* 34:67–86
52. Regis RG (2014) Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Eng Optim* 46(2):218–243
53. Regis RG (2014) Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions. *IEEE Trans Evol Comput* 18(3):326–347
54. Ren Z, Pang B, Liang Y, Chen A, Zhang Y (2018) Surrogate model assisted cooperative coevolution for large scale optimization. [arXiv:1802.09746](https://arxiv.org/abs/1802.09746) (arXiv preprint)
55. Rumelhart D, Hinton G, Williams R (1988) Learning internal representations by error propagation. In: Collins A, Smith EE (eds) *Readings in cognitive science*. Morgan Kaufmann, Burlington, pp 399–421. <https://doi.org/10.1016/B978-1-4832-1446-7.50035-2>
56. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4(4):409–423. <http://www.jstor.org/stable/2245858>
57. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N (2016) Taking the human out of the loop: a review of Bayesian optimization. *Proc IEEE* 104(1):148–175. <https://doi.org/10.1109/JPROC.2015.2494218>
58. Sun C, Jin Y, Cheng R, Ding J, Zeng J (2017) Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Trans Evol Comput* 21(4):644–660
59. Sun C, Jin Y, Zeng J, Yu Y (2015) A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Comput* 19(6):1461–1475
60. Sun C, Zeng J, Pan J, Xue S, Jin Y (2013) A new fitness estimation strategy for particle swarm optimization. *Inf Sci* 221:355–370
61. Tang Y, Chen J, Wei J (2013) A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions. *Eng Optim* 45(5):557–576
62. Tenne Y, Armfield SW (2009) A framework for memetic optimization using variable global and local surrogate models. *Soft Comput* 13(8–9):781
63. Viana FA, Haftka RT, Watson LT (2013) Efficient global optimization algorithm assisted by multiple surrogate techniques. *J Glob Optim* 56(2):669–689
64. Vincenzi L, Gambarelli P (2017) A proper infill sampling strategy for improving the speed performance of a surrogate-assisted evolutionary algorithm. *Comput Struct* 178:58–70
65. Wang H, Jin Y, Doherty J (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. *IEEE Trans Cybern* 47(9):2664–2677
66. Wang H, Jin Y, Sun C, Doherty J (2019) Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Trans Evol Comput* 23(2):203–216. <https://doi.org/10.1109/TEVC.2018.2834881>
67. Wang Y, Yin D, Yang S, Sun G (2019) Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. *IEEE Trans Cybern* 49(5):1642–1656. <https://doi.org/10.1109/TCYB.2018.2809430>
68. Ye F, Wang H, Li G (2017) Variable stiffness composite material design by using support vector regression assisted efficient global optimization method. *Struct Multidiscip Optim* 56(1):203–219
69. Yu H, Tan Y, Sun C, Zeng J (2017) Clustering-based evolution control for surrogate-assisted particle swarm optimization. In: 2017 IEEE congress on evolutionary computation (CEC), pp 503–508. IEEE
70. Yu H, Tan Y, Sun C, Zeng J (2019) A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization. *Knowl Based Syst* 163:14–25. <https://doi.org/10.1016/j.knosys.2018.08.010>
71. Yu H, Tan Y, Sun C, Zeng J, Jin Y (2016) An adaptive model selection strategy for surrogate-assisted particle swarm optimization algorithm. In: 2016 IEEE symposium series on computational intelligence (SSCI), pp 1–8. IEEE
72. Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. *Inf Sci* 454:59–72
73. Zhou Z, Ong YS, Lim MH, Lee BS (2007) Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Comput* 11(10):957–971
74. Zhou Z, Ong YS, Nair PB, Keane AJ, Lum KY (2007) Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans Syst Man Cybern Part C* 37(1):66–76

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.