This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

A Novel Evolutionary Sampling Assisted Optimization Method for High Dimensional Expensive Problems

Xinjing Wang, G. Gary Wang, Baowei Song, Peng Wang, and Yang Wang

***Abstract*—Surrogate assisted Evolutionary Algorithms (SAEAs) are promising methods for solving high dimensional expensive problems. The basic idea of SAEAs is the integration of nature-inspired searching ability of evolutionary algorithms (EA) and prediction ability of surrogate models. This paper proposes a novel Evolutionary Sampling Assisted Optimization (ESAO) method which combines the two abilities to consider global exploration and local exploitation. Differential Evolution (DE) is employed to generate offspring using mutation and crossover operators. A global RBF surrogate model is built for prescreening of the offspring's objective function values and identifying the best one, which will be evaluated with the true function. The best offspring will replace its parent's position in the population if its function value is smaller than that of its parent. A local surrogate model is then built with selected current best solutions. An optimizer is applied to find the optimum of the local model. The optimal solution is then evaluated with the true function. Besides, a better point found in the local search will be added into the population in the global search. Global and local searches will alternate if one search cannot lead to a better solution. Comprehensive analysis is conducted to study the mechanism of ESAO and insights are gained on different local surrogates. The proposed algorithm is compared with two state-of-the-art SAEAs on a series of high dimensional problems and results show that ESAO behaves better both in effectiveness and robustness on most of the test problems. Besides, ESAO is applied to an airfoil optimization problem to show its effectiveness.**

***Index Terms*—Evolutionary algorithms, evolutionary sampling, high dimensional expensive problems, surrogate models.**

This work was supported in part by National Natural Science Foundation of China Grant 51375389, China Grant 51875466, China Grant 51805436, in part by China Scholarship Council. *(Corresponding author: Peng Wang)*

X. Wang is with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China. (e-mail: wangxinjing0213@hotmail.com)

G. G. Wang is with School of Mechatronic Systems Engineering, Simon Fraser University, BC, Canada. (e-mail: Gary_Wang@sfu.ca)

B. Song is with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China. (e-mail: hanghai@nwpu.edu.cn)

P. Wang is with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China. (e-mail: wangpeng305@nwpu.edu.cn)

Y. Wang is with the School of Management, Northwestern Polytechnical University, Xi'an 710072, China. (e-mail: yangw@nwpu.edu.cn)

I. INTRODUCTION

REPLACING experiments and empirical formulas with computer simulations, such as Computational Fluid Dynamics (CFD), Finite Element Analysis (FEA), are becoming more and more popular in engineering practice. Simulation-based design and optimization aims at dealing with optimization problems which need expensive computer simulations. These kinds of problems, which usually do not have function expression, are called black-box problems. Since each simulation is often time-consuming (several hours, even days), it is preferred to have fewer number of function evaluations required for optimization.

Surrogate models relieve the computational expense by replacing the simulations with an approximation model. Comparing to simulation, the cost of predicting a function value from a surrogate is assumed negligible. Widely used surrogate models include Polynomial Response Surface (PRS) [1], Kriging [2], Radial Basis Functions (RBF) [3], Support Vector Regression (SVR) [4] and Multivariate Adaptive Regression Splines (MARS) [5], and so on. The performance of different surrogate models under multiple criteria has been studied [6]. For simulation-based optimization, promising sample points are usually chosen iteratively to gradually update the surrogate. The process repeats until a stopping criterion is met. Strategies of determining promising points are called sampling strategies. A good sampling strategy should consider both global exploration and local exploitation [7]. A few popular sampling strategies including maximizing the expected improvement (EI) [8], [9], maximizing the probability of improvement function (PI) [9], and minimizing the lower confidence bound (LCB) [9], have been proved effective. In recent years, more methods have been proposed with novel ideas [10]-[15]. Besides, some researchers are interested in choosing more than one sample point per cycle [16]-[18]. These newly picked points could be simulated in parallel. If parallel computing resources are available, the number of iterations could be greatly reduced. Haftka et al. [19] stated that methods that provide easy parallelization or methods that rely on population of designs for diversity deserve more attention.

However, most of these aforementioned strategies deal with small-scale problems whose dimensions are smaller than 15 [20]. For medium-scale (20-50 design variables) and large-scale problems (>50 design variables), these methods will

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

not work well because it is nearly impossible to build an accurate surrogate model for such high dimensional problems with limited number of points. Shan and Wang [21] published a survey of modeling and optimization strategies for solving high dimensional problems. They integrated RBF with high dimensional model representation (HDMR) to develop a new model, RBF-HDMR [22]. Test results confirmed the efficiency and capability of RBF-HDMR for medium-scale problems. Recently, Wu et al. [23] proposed a Partial Metamodel-based Optimization (PMO) method using RBF-HDMR and results showed that PMO performs better than optimizing a complete RBF-HDMR. Decomposition that reformulates a problem into a set of sub-problems of smaller scale is also a promising method for solving high dimensional problems [24], [25].

Another popular trend is to combine EA with surrogate models to deal with computationally expensive problems, which are called Surrogate-assisted Evolutionary Algorithms (SAEAs) [26]-[29]. Some SAEAs focus on low-dimensional problems [30]]-[[32]. For example, Tang et al. proposed a surrogate-based PSO algorithm for expensive black-box problems and applied the algorithm to several engineering problems [30]. Vincenzi et al. proposed an improved SAEA, which combines the Differential Evolution with a quadratic surrogate and an infill sampling strategy. The performance of the method was carried out on benchmark functions and engineering applications [31], [32]. Some researchers focus on applying SAEAs to deal with high dimensional expensive problems [33]-[36]. Regis [20] built a global RBF surrogate model to predict promising offspring of PSO and conducted local search in a sub-region of the current best solution. Regis also proposed a SAEA which can deal with inequality constraints [37]. Liu et al. [38] utilized Gaussian process to predict offspring generated by current best samples and dimension reduction techniques were also used for tackling 50 variables problems. Wang et al. [39] applied an ensemble of surrogates to build a global model and PSO was used to find the optimum of the model. The most uncertain solution was also searched at the same time. Besides, a local surrogate model was applied to accelerate the exploitation. Their results are outstanding on medium-scale test functions with limited number of function evaluations. Sun et al. combined two PSO methods to solve 50 dimensional (50D) and 100D problems and attempted to optimize 200D problems [40]. Yu et at.[41] proposed an surrogate-assisted hierarchical PSO (SHPSO) algorithm for high dimensional expensive problems. SHPSO uses a local RBF network to guide the selection of new samples. Results show that SHPSO performs better than GPEME and SA-COSO.

To tackle high dimensional expensive problems, in this study, we propose an optimization method called Evolutionary Sampling Assisted Optimization (ESAO). ESAO consists of two major parts, the global search and local search. The global search employs a global RBF model to choose the best offspring generated by evolutionary operators (mutation and crossover). The local search utilizes an optimizer to search for the optimum of the local surrogate model which is trained with a certain number of current best points. The rest of the paper is organized as follows. Section 2 briefly introduces the related techniques including DE and surrogate models. Section 3 describes the proposed ESAO in detail. In Section 4, comprehensive analysis is conducted to study the behavior of ESAO and comparison is made with other well-known SAEAs. An engineering application is also included to show the effectiveness of ESAO. Conclusions and future research are described in Section 5.

II. Related Techniques

A. Differential Evolution

As one of the most powerful stochastic real-parameter optimization algorithms, DE has been drawing the attention from all over the world since its inception in 1995 [42]. Many researchers have proposed many different variants of the basic algorithm with improved performance. In general, there are four stages of DE, initialization, mutation, crossover, and selection. Mutation and crossover make the main contributions on searching for better candidate solutions.

Assuming we have a population at current generation, $x=[x_1,x_2,...,x_n]^T$. Each component of x is an individual with dimension d, $x_i=(x_i^1,x_i^2,...,x_i^d)$. The mutation process can be expressed as

$$v_i = x_{i1} + F\cdot(x_{i2}-x_{i3}) \tag{1}$$

where x_{i1}, x_{i2} and x_{i3} are randomly chosen different individuals from the current population. F is a scalar number which typically lies in the interval [0.4, 1] [42]. There are many variants of mutation and we just adapt DE/best/1 to generate offspring in this study. For DE/best/1, the only difference from Eq. (1) is that x_{i1} is the current best solution rather than a random individual.

Crossover aims to enhance the potential diversity of the population and is conducted after mutation. There are two kinds of crossover methods, exponential and binomial. We just introduce the latter in this study. The binomial crossover formula can be expressed as

$$u_i^j = \begin{cases} v_i^j & if\,(rand_i^j \in [0,1] \le Cr \;\; or \;\; j = j_{rand}) \\ x_i^j & otherwise \end{cases} \tag{2}$$

where u_i^j is the j^{th} component of i^{th} offspring. x_i^j and v_i^j are the j^{th} component of i^{th} parent and the mutated individual. C_r is a constant between 0 and 1. *rand* indicates a uniformly distributed random number. $j_{rand}\in[1,2,...,d]$ is a randomly chosen index that ensures u_i has at least one component of v_i. Readers who are interested in the comprehensive introduction of DE can refer to [42].

B. Surrogate models

Kriging and RBF are two most widely studied surrogate models. The main advantage of Kriging is that Kriging can yield an estimation of mean square errors (MSE). The accuracy of Kriging can be improved through adding samples in the region where MSE is large. A disadvantage of Kriging is that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

training of the model is time consuming when the number of samples is large. RBF is another promising modeling method with high efficiency and accuracy. As many publications have introduced the two methods and their variants, we won't report them in detail. Readers who are interested in the two methods can refer to [7].

Previous studies showed that different surrogate models suit for different problems [43] and guidelines on model selection were also given. However, the dimensions of the benchmark functions they dealt with are smaller than 16. Their conclusions may not be suitable for high dimensional problems. Besides, engineers cannot choose a suitable surrogate model without knowing the property of an engineering problem beforehand. Some researchers proposed to construct an ensemble of surrogates [43], [44]. In this way, the usage of the worst surrogate model can be avoided and sometimes the accuracy of an ensemble is higher than that of all the individual models.

III. Proposed ESAO method

ESAO contains two parts, the global search and the local search. The global search aims to search for the entire design space with the help of DE operators. The generated offspring are expected to provide a good coverage of current promising regions and unexplored regions. A RBF global model is built using all the available sample points to predict the responses of the offspring. The most promising offspring will be evaluated using the true function. In this study, RBF is used as the global surrogate model since it is time-consuming to train a Kriging model when the number of sample points is large. It is reported that it will take several minutes to train a Kriging model, when hundreds of points are involved [38]. The local search aims to accelerate the searching in promising sub-regions. A local surrogate model is built with a certain number of current best sample points. An optimizer is used to search for the optimum of the local model. The optimum will be evaluated with the true function. As both Kriging and RBF can be employed for the local surrogate models, investigations about which model (Kriging and RBF) is more effective for the local model should be done. We expect the results give some insights about the selection of surrogate models when dealing with high dimensional problems. Global Search Strategy

Algorithm 1 shows the pseudo code of the global search. Database stores all the sample points and their function values. At the beginning of the process, Optimal Latin Hypercube Sampling (OLHS) [45] is used to generate the initial population. Mutation and crossover are conducted on the current population to generate a new population of the same size. All sample points in the database are used to train a RBF global model, f_g, which gives predictions of the offspring. The offspring (x_g) with the lowest prediction will be evaluated with the true function. If the function value of x_g is better than that of its parent, $f(x_{gp})$, x_g will replace its parent's position. If the function value of x_g is better than that of the current best sample (x_b), x_g substitutes x_b. As x_g is an expensive sample point, it will be added into the database and the global RBF will be updated. If the global search finds a better solution, this process will continue. Otherwise, the optimization will turn to the local search.

Algorithm 1 *Pseudo Code of the Global Search*

1: Database, including sample points and their function values; Current population; Present best solution (x_b);
2: **Repeat**
3: Generate offspring using mutation and crossover operators of DE according to Eqs. (1) and (2) on current population;
4: Use all the points in the database to build the RBF model, f_g;
5: Obtain the predictions of all the offspring using f_g;
6: Choose the offspring with the lowest prediction, x_g and evaluate its true function value, $f(x_g)$; Save x_g and $f(x_g)$ into the database;
7: **if** $f(x_g) < f(x_{gp})$ **then**
8: Replace x_{gp} with x_g and update the population;
9: **end if**
10: **if** $f(x_g) < f(x_b)$ **then**
11: Replace x_b with x_g;
12: **end if**
13: **Until**: Step 10 is not true.

A. Local Search Strategy

The local search aims to search for the optimum of the local surrogate model in sub-regions and its pseudo code is shown in Algorithm 2. We use τ best sample points from the database to train the local surrogate model. The τ best sample points will form a sub-space and a global optimizer (GA, DE, PSO) or multi-start searching algorithm [46] could be used to find the optimum of the local model. In this research, DE is chosen as the optimizer of the surrogate model, since DE operators are also used in the global search. However, it should be noted that the optimizer DE is not the same as the one in the global search. They are two independent instances. The DE used in the global search just generates offspring with evolutionary operators, while the DE used in the local search serves as an optimizer. The optimum found by the optimizer will be accepted and its true function value is calculated. If the newly added sample has a lower function value than that of the current best solution, it will replace x_b and be added to the population in the global search. It has been noted that the population size in global search will not increase and a better solution will just replace its parent's position. When the local search finds a better solution than the current best solution, the new best solution will be added into the population in the global search. In this way, the size of the population will increase. The local search process continues when a better solution is found. As the number of samples for local surrogate models is stable, one of these local points will be eliminated and a new one joins the group. In this way, samples with lower function values will concentrate on smaller and smaller regions. If no better sample is found, the optimization will switch to the global search.

To reduce the computation burden, τ is set as $2\times d$, where d is the dimension of the problem. If Kriging is set as the local surrogate model, it will take too much time for the training with $3\times d$ best samples when d is 100 or higher; while we do not think it is enough to train the local surrogate model with only d best sample points when d is 20. The lower and upper bounds of each dimension are the minimum and maximum of all the $2\times d$ samples, respectively. Eq. (3) gives the formula to identify the sub-region.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

$$sub\_space = \begin{cases} lb_i = \min(x_j^i) \\ ub_i = \max(x_j^i) \end{cases} \tag{3}$$
$$i = 1, 2, ..., d \quad j = 1, 2, ..., 2d$$

where lb_i and ub_i stand for the i^{th} lower and upper bounds of x_i, respectively. x_j represents one of the $2\times d$ number of samples.

Algorithm 2 *Pseudo Code of the Local Search*

1: Database, including samples and function values; Present best sample (x_b).
2: **Repeat**
3: Select τ best sample points from the database to train the local model (f_l);
3: Run optimizer to find the optimum of the model;
4: Evaluate x_l using the true function and save x_l and $f(x_l)$ into the database;
5: **if** f $f(x_l) < f(x_b)$ **then**
6: Replace x_b with x_l;
7: Add x_l to the population in global search;
8: **end if**
9: **Until**: Step 5 is not true.

B. Framework of the Proposed ESAO

Fig. 1 shows the framework of ESAO. The solid arrows stand for the flows of the algorithm, while the dotted arrows represent the data flows. The explanations of the five data flows are given below.

① When the function value of the best offspring is better than its parent, the parent will be replaced by the offspring. When a local search finds a better solution than the current best result, the better solution will be added into the population in the global search.

② All sample points and their function values in the database will be used to train the global RBF model.

③ Initial samples and their function values are generated using OLHS and they will be transferred into the database. Note that the initial samples form the initial population in the global search and this data flow just occurs once.

④ The best offspring picked by the global RBF model and the optimum found through the local surrogate are evaluated using the true function. The sample points and their function values will be saved into the database.

⑤ τ best sample points in the database will be picked to train the local surrogate model.

It should be noted that the design space is normalized into [0, 1]. For the framework, initial samples are obtained using OLHS and their function values are calculated with the true function. Then, the global and local searches are conducted. It can be seen that the global or local searches will continue running if they find a better solution. The two parts alternate when no progress is made. Both the global and local searches just choose one sample per cycle and the function value of the sample is evaluated with the true function. The newly chosen sample and its function value will be added to the database regardless if the function value is better than that of the current best sample. The sample points that are used in global and local searches will be updated correspondingly.

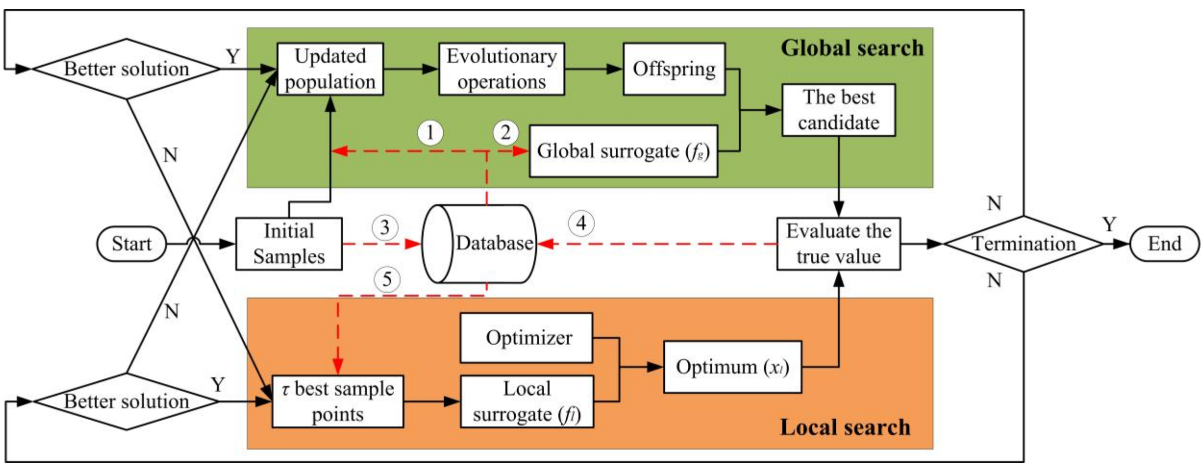

Fig 1 Framework of the proposed ESAO

IV. Results and Discussion

To investigate the effectiveness of ESAO, comparisons are conducted with state-of-the-art algorithms on various types of high dimensional problems. First, test problems and parameters used in ESAO will be explained. Second, comprehensive analysis is conducted for ESAO, including the effect of adding better samples into the population, and the contributions of the global and local searches in finding the optimum. We also compared the effect when different surrogate models act as the local model. Finally, ESAO is compared with two well-known SAEAs on the test problems. An airfoil design problem is followed to show the effectiveness of ESAO in engineering application.

TABLE 1 Properties of the test problems

Problem	*d*	Optimum	Property
Ellipsoid	20, 30, 50, 100	0	Unimodal
Rosenbrock	20, 30, 50, 100	0	Multimodal
Ackley	20, 30, 50, 100	0	Multimodal
Griewank	20, 30, 50, 100	0	Multimodal
Shifted Rotated Rastrigin	30, 50, 100	-330	Very complicated multimodal
Rotated hybrid Composition Function	30, 50, 100	10	Very complicated multimodal

A. Experiment Settings

The SURROGATES toolbox [47] was used for implementation of ESAO. Specifically, Kriging and DE are adopted from the toolbox. When Kriging is the local surrogate model, Gaussian correlation function is used. Multiquadric kernel function is adopted in RBF. In the global search, both *F* and C_r are set as 0.8 according to [38]. The DE used in the local search adopts DE/best/1 strategy, 150 population members, 200 maximum number of generations, 0.8 for both F and C_r . In ESAO, the initial number of sample points is set as 100 for 20D and 30D problems and 200 for 50D and 100D problems. The optimization terminates when the total number of function evaluations (NFE) reaches 1000 for all the test problems. Six series of problems with different dimensions and complexities are included and totally 22 problems are involved. The basic information of the problems is given in Table 1. The first four problems have four kinds of dimensions, 20, 30, 50 and 100 and the last two have three kinds of dimensions, 30, 50 and 100. It should be noted that a simplified name is used for easy understanding of a specific problem. For example, 'E20' means this is a 20D Ellipsoid problem. SRR is the abbreviation of Shifted Rotated Rastrigin problem and RHC is the abbreviation of Rotated Hybrid Composition problem. Readers who are interested in the function expressions of the test problems can refer to [38]. For all test problems, 30 independent runs are conducted to obtain statistical results.

ESAO is compared with three well-known SAEAs, GPEME [38], SA-COSO [40] and SHPSO [41]. GPEME was tested with 20D, 30D and 50D problems; SA-COSO was tested with 50D, 100D and 200D problems; and SHPSO was tested with 30D, 50D and 100D problems. The three methods are recently published promising algorithms, which deal with high dimensional problems with fewer NFE (<1000).

TABLE 2 STATISTICAL RESULTS OF 22 TEST PROBLEMS OBTAINED BY EASO

Problems	Best	Worst	Mean	Std.
E20	1.2905E-5	2.52E-3	1.8099E-4	4.6815E-4
E30	8.6562E-5	0.2782	0.02747	0.06964
E50	0.1646	2.2644	0.7395	0.5549
E100	1102.3190	1538.8248	1282.8950	134.3897
R20	12.7925	19.2266	15.1618	1.6289
R30	22.1577	29.4037	25.0363	1.5701
R50	43.1221	49.2492	47.3914	1.7118
R100	521.2047	673.2419	578.8427	44.7671
A20	2.5980	11.2992	6.8648	3.2586
A30	0.0780	3.9096	2.5213	0.8396
A50	1.0571	2.4326	1.4311	0.2491
A100	9.9664	10.7321	10.3640	0.2113
G20	0.8645	1.0197	0.9719	0.03910
G30	0.7860	1.0221	0.9534	0.05037
G50	0.8518	1.0207	0.9404	0.04209
G100	47.3461	69.2247	57.3417	5.8387
SRR30	-35.7804	90.3323	6.3250	26.4772
SRR50	116.2469	289.0872	198.6141	45.8253
SRR100	662.6292	758.8105	713.4680	26.4540
RHC30	923.3490	953.8883	931.6703	8.9417
RHC50	940.9899	1049.9114	975.3207	37.1101
RHC100	1321.8061	1427.1364	1372.4218	27.5390

B. *Behavior study of ESAO*

According to our practice, ESAO will have better performance when RBF is the local surrogate model. Therefore, the following analysis is based on the ESAO with RBF being the local model. Table 2 shows the results of ESAO on 22 test problems. It can be seen that ESAO can find very good results for 20D, 30D and 50D problems, especially for Ellipsoid, rosenbrock, and Griewank problems. Ellipsoid is a relatively simple problem, the optimum found by ESAO of 20D and 30D problems are very close to the true optima. Although Griewank problem is multimodal, ESAO finds very good results for 20D, 30D and 50D problems. A common understanding is that the optimum found on higher dimensional problem is worse than that of lower dimensional problem with the same NFE. This phenomenon appears in Ellipsoid and Rosenbrock problems. However, Ackley and Griewank series problems are exceptions. When the dimension of the problem increases, the optima found by ESAO become smaller. According to Locatelli's research [48], it is easier to identify the global region of Griewank problem when the dimension increases, since the effect of the product of cosine components (the second term of Griewank function) can be neglected. The results mean that ESAO could identify the promising regions efficiently for Griewank problem. ESAO cannot find very good results for SRR and RHC series problems with limited NFE, since they are very complicated multimodal. When it comes to 100D problems, the results are not as well as those in lower dimensional problems. The standard deviations (Std.) are also provided, which can reflect the robustness of ESAO. It can be seen that the Std. are quite small for 20D, 30D and 50D problems, but a bit larger for 100D problems.

TABLE 3 COMPARISON OF THE RESULTS WITH OR WITHOUT INCREASING THE POPULATION SIZE

Problems	ESAO		ESAO-N	
	Mean	Std.	Mean	Std.
E20	**1.8099E-4**	**4.6815E-4**	5.1569	4.2578
E30	**0.02747**	**0.06964**	10.1470	4.9803
E50	**0.7395**	**0.5549**	16.2176	13.0671
E100	**1282.8950**	**134.3897**	1305.6326	142.7622
R20	**15.1618**	**1.6289**	64.6682	26.3719
R30	**25.0363**	**1.5701**	110.9915	36.1576
R50	**47.3914**	**1.7118**	73.3485	17.4618
R100	**578.8427**	**44.7671**	652.3402	56.7690
A20	**6.8648**	3.2586	9.7798	**0.9230**
A30	**2.5213**	**0.8396**	3.5972	2.8520
A50	**1.4311**	0.2491	1.5830	**0.1676**
A100	**10.3640**	**0.2113**	10.6266	0.2439
G20	0.9719	0.03910	**0.9631**	0.0479
G30	**0.9534**	0.05037	0.9620	**0.0402**
G50	**0.9404**	0.04209	0.9542	**0.0405**
G100	**57.3417**	5.8387	65.7568	**4.5053**
SRR30	**6.3250**	**26.4772**	18.2936	32.6004
SRR50	**198.6141**	45.8253	218.6177	**35.5651**
SRR100	**713.4680**	**26.4540**	736.0916	51.9083
RHC30	**931.6703**	**8.9417**	967.7603	27.8783
RHC50	**975.3207**	37.1101	1021.4304	**30.3683**
RHC100	**1372.4218**	**27.5390**	1385.6125	34.5613

For ESAO, the global search combines the prediction ability of the RBF model and the search ability of the evolutionary algorithm. The global RBF model adopts all the samples in the database for its construction. The search ability embodies in the generation of offspring using evolutionary operators (mutation and crossover). A new idea that could enhance the search ability of the global search is to increase the population size with better solutions. Since a better solution found in the global search will just replace its parent's position, a better solution found in the local search will also be added into the population. In this way, the size of population in the global search will increase as iteration goes on. It should be noted that the mutation strategy, DE/best/1 in ESAO uses the current best solution to guide the generation of the offspring. When the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

better solution is found by the local search, the mutation strategy will generate offspring based on the new best point in the global search.

Comparison is conducted to validate the effect of adding better solutions into the population. The results are shown in Table 3. In Table 3, ESAO-N is the variant without better samples added into the population. The better values are boldfaced. It can be seen clearly that the average best values of ESAO are smaller than those of ESAO-N on 21 out of 22 problems and the values are significant better on 20D, 30D and 50D problems. This is because the global search is enhanced with more and more better samples added into the population. The global search could generate more promising offspring around the newest best solution. In this way, the design could be explored more effectively. However, it should be noted that the average best values are not that better on 100D problems because it is becoming more difficult for global search to find better solutions in such a high dimensional space even if better samples are added. On the other side, the Std. of ESAO are much smaller than those of ESAO-N on most of the problems. If not, the values of the two methods are very close. This means the proposed idea also increases the robustness of ESAO. From the comparison, we can see that ESAO can find better solutions with stronger robustness when adding better samples found in the local search into the population of the global search.

TABLE 4 AVERAGE NFE AND NTI OF THE GLOBAL AND LOCAL SEARCH

Problems	Global search		Local search	
	NFE	NTI	NFE	NTI
E20	393.5	6.2	506.5	119.7
E30	377.8	2.3	522.2	147.2
E50	349.9	0.5	450.1	101.2
E100	384.2	0.8	415.8	32.7
R20	367.7	40.0	532.4	205.2
R30	347.9	23.6	552.1	228.4
R50	345.5	0.5	454.5	110.0
R100	384.7	0.9	415.3	32.2
A20	445.4	0.2	454.7	10.0
A30	434.9	0.3	465.1	30.9
A50	373.2	0.5	426.8	54.7
A100	384.9	1.5	415.1	32.3
G20	436.7	0.8	463.3	27.9
G30	431.5	0.7	468.5	38.1
G50	365.3	1.3	434.7	71.0
G100	381.7	1.8	418.3	38.9
SRR30	448.8	2.0	451.2	5.0
SRR50	396.6	1.2	403.4	8.5
SRR100	392.7	0.3	407.3	15.4
RHC30	421.5	6.4	478.5	64.0
RHC50	366.6	0.7	433.4	68.2
RHC100	397.1	0.5	402.9	6.6

An interesting question is that for global and local searches, which part makes more contributions in finding better solutions? Besides, how many sample points are picked by the global search and by the local search? Table 4 shows the average NFE and number of true improvements (NTI) of the two searches on all the test problems. It should be noted that global and local searches will pick 900 points for 20D and 30D problems and 800 points for 50D and 100D problems, with the initial number of samples being 100 and 200 respectively. For example, the NFE of global and local searches are 393.5 and 506.5, respectively for E20, while the NTI of the two searches are 6.2 and 119.7.

The NFE of the global and local searches represents how many expensive function evaluations are called for one search. It can be seen that the NFE of local search is always larger than that of the global search, except for SRR series problems. The NFE of the local search is usually larger than 450 for 20D and 30D problems and larger than 400 for 50D and 100D problems. The second phenomenon is that NTI found by the local search is much larger than that of the global search. This is reasonable because the goal of the global search is exploration and the local search aims at exploitation. The local search tries to find a better solution in sub-regions, while the global search makes sure that ESAO does not get trapped into a local optimum. The third observation is that NTI becomes smaller with the increase of dimension for both global and local searches. This indicates that it is more difficult to find a better solution in higher dimensional space. The NTI in the global search also reflects the results in Table 3. As the global search cannot contribute much in exploitation, adding better solutions into the population will not have obvious effect for 100D problems.

TABLE 5 COMPARISON OF THE RESULTS WITH DIFFERENT LOCAL MODELING STRATEGIES

Problems	ESAO-R		ESAO-K		
	Mean	Std.	Mean	Std.	W-test
E20	**1.8099E-4**	**4.6815E-4**	0.06718	0.1409	+
E30	**0.02747**	**0.06964**	21.2342	13.0201	+
E50	**0.7395**	**0.5549**	1584.7371	177.0219	+
E100	**1282.8950**	**134.3897**	9944.5869	1123.2832	+
R20	**15.1618**	**1.6289**	23.7772	16.9419	+
R30	**25.0363**	**1.5701**	75.1889	38.3428	+
R50	**47.3914**	**1.7118**	1784.6183	353.5829	+
R100	**578.8427**	**44.7671**	5915.7172	733.1805	+
A20	**6.8648**	3.2586	12.5662	2.5358	+
A30	**2.5213**	**0.8396**	13.3111	1.6188	+
A50	**1.4311**	**0.2491**	17.4773	0.4557	+
A100	**10.3640**	0.2113	18.6595	**0.1630**	+
G20	**0.9719**	**0.03910**	1.4943	0.8351	+
G30	**0.9534**	**0.05037**	3.7251	1.4911	+
G50	**0.9404**	**0.04209**	240.2074	31.2303	+
G100	**57.3417**	**5.8387**	699.1255	53.0977	+
SRR30	6.3250	**26.4772**	**-27.1770**	44.0855	-
SRR50	**198.6141**	45.8253	312.9193	**35.9091**	+
SRR100	**713.4680**	**26.4540**	1470.1043	118.0472	+
RHC30	**931.6703**	**8.9417**	976.2029	37.0014	+
RHC50	**975.3207**	**37.1101**	1033.6277	**33.9043**	+
RHC100	**1372.4218**	27.5390	1453.0480	**22.8589**	+

As discussed in the local search, both Kriging and RBF can be used to build the local model, since only a small number of points is involved. Investigation should be conducted to figure out which one can lead to better results for ESAO. ESAO with Kriging and RBF as the local model are called ESAO-K and ESAO-R, respectively. Other settings remain the same. Besides the average best values and Std., Wilcoxon rank sum tests at a significant level of $\alpha = 0.05$, are also provided, where '≈' indicates no obvious difference between ESAO-R and ESAO-K, '+' means ESAO-R significantly outperform ESAO-K, '-' means ESAO-K outperforms ESAO-R.

The comparison results are shown in Table 5. It can be seen that the results obtained by ESAO-R are much better than those of ESAO-K, except for SRR30. The differences of the two variants become larger with the increase of dimension. This indicates that Kriging is not competitive as a local surrogate model. When DE searches for the optimum of Kriging, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

optimal solution is actually not a promising result. The reason that Kriging performs much worse than RBF may be that the parameters for Kriging's maximum likelihood estimation are at a local optimum, and the cost to obtain the global optimum of these parameters would be prohibitive. ESAO-K only behaves better on SRR30 problem, which could be attributed to adaptability of Kriging for this problem. It indicates that Kriging is suitable for some problems, even if RBF can lead to better results on most of the problems. As the results of ESAO-R are better, we will use the results of ESAO-R to compare with the other two methods, GPEME and SA-COSO. ESAO that appears in the whole paper is equivalent to ESAO-R.

TABLE 6 COMPARISON OF ESAO AND GPEME ON 20D, 30D AND 50D PROBLEMS

Problems	ESAO		GPEME	
	Mean	Std.	Mean	Std.
E20	1.8099E-04	4.6815E-04	**1.30eE-05**	**2.18E-05**
E30	**0.02747**	0.06964	0.0762	**0.0401**
E50	**0.7395**	**0.5549**	221.0774	81.6123
R20	**15.1618**	**1.6289**	22.4278	18.7946
R30	**25.0363**	**1.5701**	46.1773	25.5199
R50	**47.3914**	**1.7118**	258.2787	80.1877
A20	6.8648	3.2586	**0.199**	**0.5771**
A30	**2.5213**	**0.8396**	3.0105	0.925
A50	**1.4311**	**0.2491**	13.2327	1.5846
G20	0.9719	**0.03910**	**0.0307**	0.0682
G30	**0.9534**	**0.05037**	0.9969	0.108
G50	**0.9404**	**0.04209**	36.6459	13.1755
SRR30	6.3250	**26.4772**	**-21.861**	36.4492
RHC30	**931.6703**	**8.9417**	958.5939	25.6946

C. *Comparison with other algorithms*

The results of ESAO are compared with GPEME on medium-scale problems (20D, 30D, and 50D) [38]. GPEME uses 50 current best solutions to generate offspring. A local Gaussian process (Kriging) is built using 100 current best samples to prescreen the offspring. The one with the lowest prediction will be added to a database. It can be seen that GPEME is a local search method. Its basic assumption is that the current best solutions will focus on smaller and smaller sub-regions with better solutions found. A dimension reduction (DR) method is also employed to help the Gaussian process to find promising offspring. Results showed that DR is helpful to find better results in 50D problems.

Table 6 shows the average best values and standard deviation found by ESAO and GPEME on 14 problems. It can be seen that ESAO outperforms GPEME on 10 (boldfaced data) out of 14 problems. GPEME performs better on E20, A20, G20 and SRR30. For most of 30D and 50D problems, ESAO achieves better results. It is easy to understand this phenomenon since GPEME just utilizes best solutions to generate offspring. The offspring and the Gaussian process will gradually concentrate on smaller sub-regions. The mutation and crossover of current best samples may not help GPEME jump out of the local region in higher dimensional space. Besides, a local Kriging used by GPEME cannot effectively predict the offspring for higher dimensional problems. ESAO strictly generates offspring according to DE operations. The offspring can exploit promising sub-regions and explore larger space. There are higher possibilities that the global search finds unexplored region. Besides, the local search searches for the best region to accelerate the convergence. From the point of robustness, it can be seen that Std. follows the same trend with the average best values. If ESAO finds a smaller best value for one problem, its Std. is also smaller and vice versa with exception for E30, G20 and SRR30. From the comparison on 20-50D problems, we can see that ESAO outperforms GPEME on most of the problems.

TABLE 7 COMPARISON OF ESAO AND SA-COSO ON 50D, 100D AND 200D PROBLEMS

Problems	ESAO		SA-COSO	
	Mean	Std.	Mean	Std.
E50	**0.7395**	**0.5549**	51.475	16.246
E100	1282.8950	**134.3897**	**1033.2**	317.18
E200	1.7616E+04	**1.1748E+03**	**1.6382E+04**	2.9811E+03
R50	**47.3914**	**1.7118**	252.58	40.744
R100	**578.8427**	**44.7671**	2714.2	117.02
R200	**4.3185E+03**	**2.8440E+02**	1.6411E+04	4.0965E+03
A50	**1.4311**	**0.2491**	8.9318	1.0668
A100	**10.3640**	**0.2113**	15.756	0.5025
A200	**14.6958**	0.2193	1.7868E+01	**2.2319E-02**
G50	**0.9404**	**0.04209**	6.0062	1.1043
G100	**57.3417**	**5.8387**	63.353	19.021
G200	**572.9036**	**36.0425**	5.7776E+02	1.0140E+02
SRR50	198.6141	45.8253	**197.16**	**30.599**
SRR100	**713.4680**	**26.4540**	1273.1	117.19
SRR200	5.3891E+03	**156.8544**	**3.9275E+03**	2.7254E+02
RHC50	**975.3207**	37.1101	1080.9	**32.859**
RHC100	1372.4218	**27.5390**	**1365.7**	30.867
RHC200	1.4564E+03	**20.4315**	**1.3473E+03**	2.4665E+01

ESAO is also compared with SA-COSO on 50D, 100D and 200D problems that are used in [40]. SA-COSO uses two surrogate assisted PSO algorithms for exploration and exploitation. Here, we also compare all the problems used in SA-COSO. It should be noted that only 200 best sample points are chosen to train the local surrogate model for 200D problems, since the dimension is very high. Table 7 shows the average best values and Std. on all the problems. It can be seen that ESAO performs better on 12 out of 18 problems, while SA-COSO behaves better on the other six. ESAO obtains better results on five 50D problems and the results are much better for the first four problems. The two methods obtain almost the same results on the last two 50D problems. ESAO also performs better for four 100D problems. It should be noted that ESAO finds much smaller values than those of SA-COSO on Rrosenbrock problem. For the 200D problems, both of the two methods perform better on three problems. However, SA-COSO seems more powerful in the last two very complicated problems. As for the robustness, the Std. values of ESAO are smaller than those of SA-COSO on most of the problems, which means that ESAO is more robust than SA-COSO. Through the comparison, we can see that ESAO is a stronger algorithm. However, when dealing with 200D problems, it is more difficult to provide reliable predictions for offspring in such a high dimensional space. The better samples found by ESAO are just from the local search.

TABLE 8 COMPARISON OF ESAO AND SHPSO ON 30D, 50D, AND 100D PROBLEMS

Problems	ESAO		SHPSO	
	Mean	Std.	Mean	Std.
E30	**0.02747**	0.06964	0.07620	**0.04010**
E50	**0.7395**	**0.5549**	4.0281	2.0599
E100	1282.8950	134.3897	**76.106**	**21.447**
R30	**25.0363**	1.5701	28.566	**0.4044**
R50	**47.3914**	**1.7118**	50.800	3.0305
R100	578.8427	44.7671	**165.59**	**26.366**
A30	2.5213	0.8396	**1.4418**	**0.7740**
A50	**1.4311**	**0.2491**	1.8389	0.5637
A100	10.3640	**0.2113**	**4.1134**	0.5925
G30	0.9534	**0.05037**	**0.9205**	0.08806
G50	**0.9404**	**0.04209**	0.9452	0.06140
G100	57.3417	5.8387	**1.0704**	**0.02049**
SRR30	6.3250	26.4772	**-92.830**	**22.544**
SRR50	198.6141	45.8253	**134.42**	**32.256**
SRR100	**713.4680**	**26.4540**	801.73	72.252
RHC30	**931.6703**	**8.9417**	939.61	9.0177
RHC50	**975.3207**	37.1101	996.60	**22.145**
RHC100	**1372.4218**	**27.5390**	1419.8	38.238

SHPSO [41] is a new surrogate-assisted evolutionary algorithm that deals with high dimensional expensive problems. A local RBF network is built with a certain number of current best samples. First, the RBFN learns the local details of the problem and a SL-PSO algorithm [49] searches for the optimum of the RBFN model. Second, when new particles are generated, RBFN can prescreen out the particles whose estimated values are smaller than their personal bests. Comparison results between ESAO and SHPSO on 18 test problems are shown in Table 8. It can be seen that ESAO performs better on ten problems, while SHPSO behaves better on the other eight. An obvious phenomenon is that ESAO find smaller function values on 30D and 50D problems (Ellipsoid, Rosenbrock, RHC), while SHPSO is superior on 100D problems. Since both ESAO and SHPSO use a local surrogate model built with the best samples, and search for the optimum of the local model, the reason that SHPSO performs much better may owe to the way of new particles generation. SHPSO only adopts the best samples to form the initial population. With the iteration goes on, the new particles could gradually aggregate in the promising region. This feature should be more effective in higher dimensional space. ESAO employs all the initial samples to form the population, the evolutionary rate of offspring may be slower. As for the Std., both of the two algorithms show robust features on nine problems. Besides, these Std. are not much different on most of the problems, expect for four 100D problems. The performance of ESAO is comparable with SHPSO on most of the problems. ESAO is more effective for 30D and 50D problems, while needs improvements for 100D problems.

D. Comparison of convergence history

Convergence history is also an important feature for an algorithm, besides average best values and standard deviation. A good algorithm can converge faster with limited NFE. Here we compare the convergence history of ESAO with other three algorithms. The x axis is the NFE and the y axis is the natural logarithm of objective values, except for Ackley series problems and SRR30. It should be noted that we do not have the original codes of GPEME, SA-COSO and SHPSO. We do not think we can reproduce their algorithms perfectly according to their papers. Therefore, we extract the data of the convergence history in the original papers. There will be some deviations to the original data, but the overall trend can be reflected for a fair comparison. It is also noted that GPEME performs better on 20D and 30D problems without dimension reduction (DR), while performs better on 50D problems with DR. We pick the better results of corresponding problems for the comparison. For SHPSO, the convergence profiles of Ellipsoid and RHC problems are based on true function values, while the other three algorithms employ the natural logarithm of the objective values. The extracted data of SHPSO on the two problems is logarithmized for the convenience of comparison.

Fig. 2 shows the convergence history of five algorithms on all testing problems. Each row of subfigures displays a kind of problem with different dimensions and each column corresponds to different problems with the same dimension. Besides, the convergence trends of ESAO-K are involved. An obvious observation is that ESAO-R and SHPSO converge faster than the other three on most of the problems. This property is promising when fewer NFE is available for expensive problems. ESAO-R performs better than SHPSO on most of 30D and 50D problems, while SHPSO does better on most of 100D problems. SHPSO shows significant advantages on convergence speed and accuracy for the first four 100D problems. ESAO-R converges faster than GPEME on all the problems, except for three 20D and one 30D problems. ESAO-R also converges faster than SA-COSO, except for two 100D problems. The reason that ESAO-R converges faster is due to the local search. Another finding is that ESAO-K converges the slowest on almost all the problems. The convergence trend corresponds to the results in Table 5, which means Kriging is not as accurate as RBF.

In general, GPEME is more suitable for medium-scale problems and SA-COSO behaves better on higher dimensional problems. SHPSO is a very promising SAEA and shows better features on 100D problems. ESAO performs better than GPEME and SA-COSO, and is comparable to SHPSO on most of the problems. In a word, ESAO is a promising method for dealing with high dimensional expensive problems.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

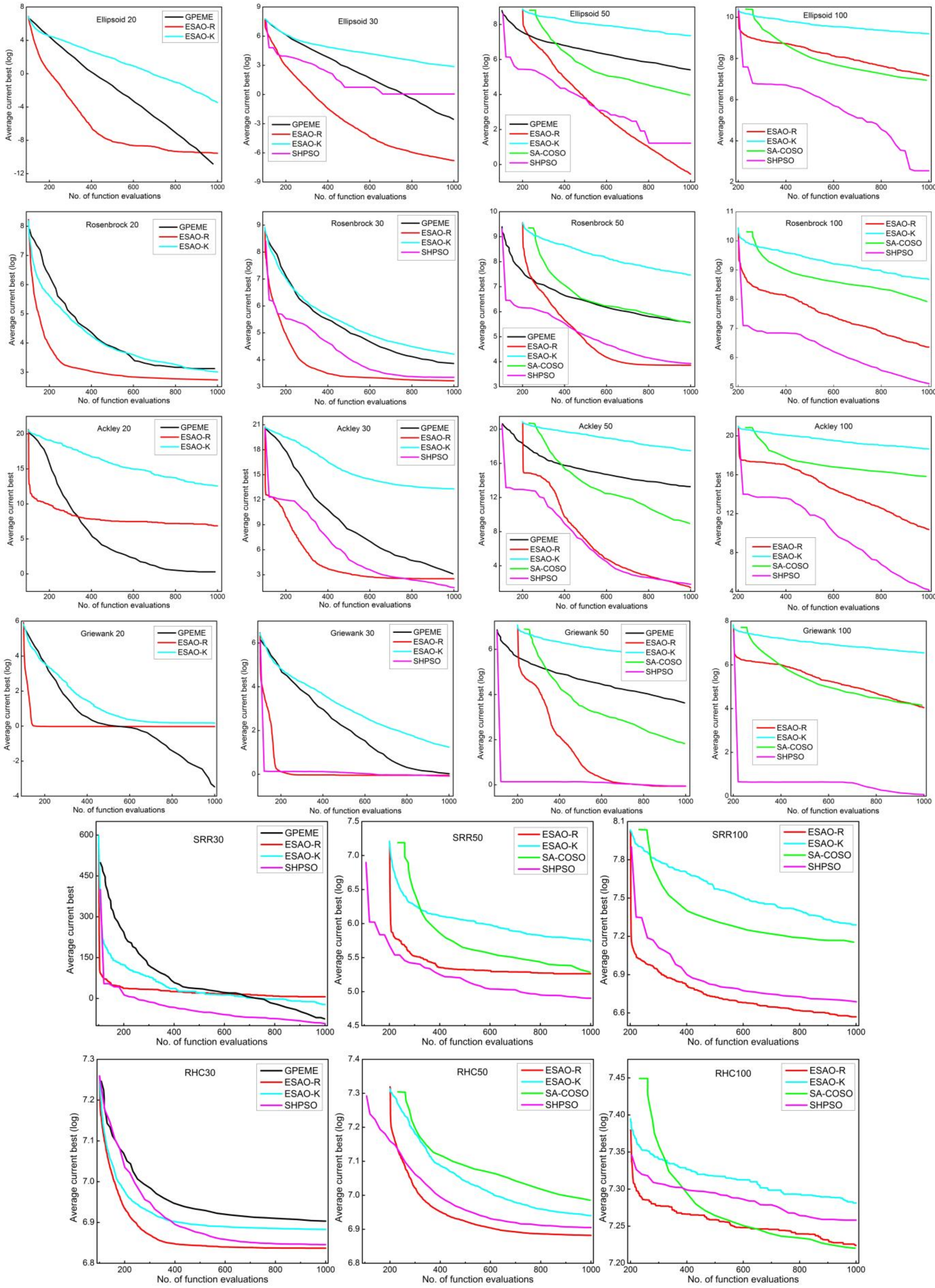

Fig. 2 Convergence history of four algorithms on all test problems

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2019.2890818, IEEE Transactions on Evolutionary Computation

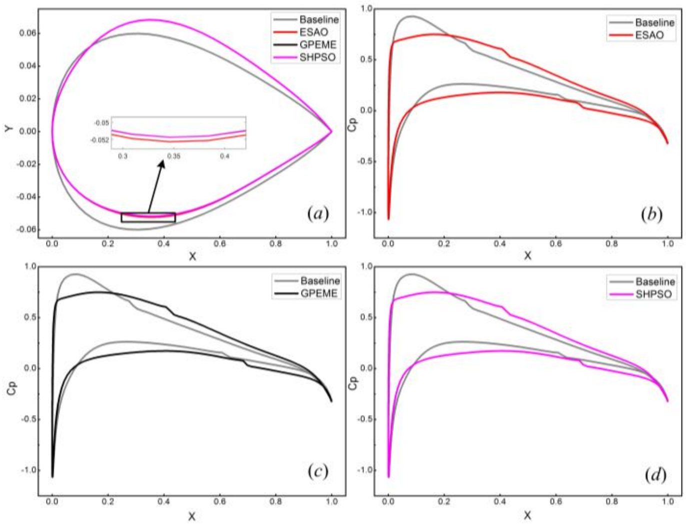

Fig. 3 Geometry and coefficient of pressure of the initial and the best optimized airfoils of three algorithms

E. Application of ESAO to an airfoil design problem

ESAO is applied to an airfoil design problem, after it has shown its effectiveness on benchmark problems. The Class-shape Transformation (CST) method is used to parameterize the airfoil geometry. NACA0012 airfoil is selected as the baseline. Six upper and lower surface coefficients are selected as the design variables. Totally, 12 variables are involved. It should be noted that the trailing edge thickness of upper and lower surfaces is set to zero. The design values of the initial airfoil are provided in Table 9. The upper and lower bounds are 120% and 80% respectively of the initial values.

TABLE 9 VALUES OF INITIAL DESIGN VARIABLES

Parameters	x_1	x_2	x_3	x_4	x_5	x_6
Values	0.1703	0.1602	0.1436	0.1664	0.1105	0.1794
Parameters	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
Values	-0.1703	-0.1602	-0.1436	-0.1664	-0.1105	-0.1794

Xfoil [50] is used to calculate the lift and drag of the airfoil. The lift-to-drag ratio (*L/D*) could be obtained accordingly. The design objective is to maximize the *L/D* with the Mach number of 0.5, angle of attack (AOA) of 2 degrees and Reynolds number of 5e6. The constraint is that the maximum thickness of the optimized airfoil is larger than that of the initial design. The optimization problem can be formulated as Eq. (4) .

$$\begin{aligned} \max \quad & L/D \\ s.t. \quad & t_0 - t_{\max} \le 0 \\ & 0.8x_i^0 \le x_i \le 1.2x_i^0 \\ & i = 1, 2, ..., 12 \end{aligned} \tag{4}$$

The settings of ESAO are as follows. The initial number of points is 50 and the total number of simulations is 300. The number of points that used to train the local surrogate model is also 50. As ESAO has not been developed to deal with constrained problems, penalty function is used to transform the constrained problem into an unconstrained one.

In addition to ESAO, GPEME and SHPSO are also included to optimize the airfoil problem. Since the problem just owns 12 design variables, some modifications on GPEME and SHPSO need to be claimed. In GPEME and SHPSO, 100 best samples are included to build the surrogate model. When the number of samples is less than 100, all of samples in the database are used. Other settings of the two algorithms follow the original methods. For each algorithm, twenty independent runs are carried out and the statistical results are given.

TABLE 10 OPTIMIZATION RESULTS OBTAINED BY THREE ALGORITHMS

Algorithm	L/D			
	Best	Worst	Mean	Std.
ESAO	80.0864	77.5431	79.2765	0.8060
GPEME	80.4762	79.4311	79.9798	0.3248
SHPSO	80.3463	76.4967	78.9243	1.1845

Table 10 shows the statistical results of the airfoil problem using ESAO, GPEME and SHPSO. All the three algorithms find better designs with constraints satisfied. It can be seen that GPEME performs the best on all the criteria. Both ESAO and SHPSO find some worse designs. GPEME also finds the largest average values, and ESAO and SHPSO hold almost the same performance. GPEME also has the smallest Std. In a word, the mean values obtained by the three algorithms are almost identical. GPEME performs the best, while ESAO and SHPSO behave almost the same.

Fig. 3 shows the best optimized geometries of the airfoil by the three algorithms and the corresponding coefficient of pressure (C_p) along the chord. It can be seen from Fig. 3(a) that three optimized airfoils are almost identical. The upper surface raises upward, while the lower surface depresses. The small differences of three optimized airfoils exist in the maximum thickness. The enlargement in Fig. 3(a) shows the maximum thickness obtained by ESAO is a little larger. Fig. 3(b)-(d) give the C_p of the baseline and optimized airfoils. It can be seen that the drag is weakened for all the optimized design and the area of the covered region by the optimized C_p become larger, which means the L/D increases. Besides, three optimized airfoils show almost the same C_p, because of the similar outlines.

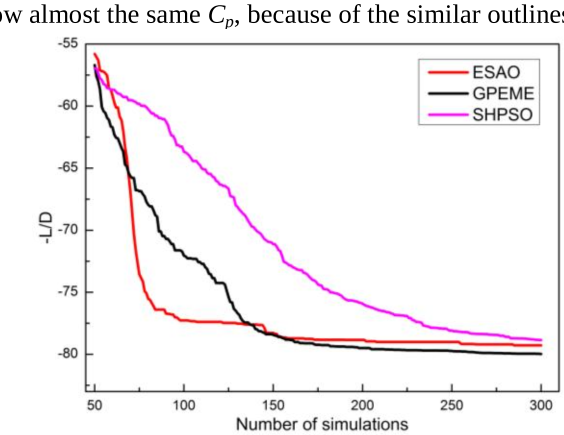

Fig. 4 Convergence history of the airfoil problem using three algorithms

Fig. 4 shows the convergence history of the *L/D* using three algorithms. It can be seen that ESAO converges the fastest at first until the number of simulations reaches 150 and the L/D does not become much better after that. GPEME finds better result continuously at first, and can find smaller result after 150 simulations. SHPSO converges the slowest among the three algorithms. In a word, the three state-of-the-art SAEAs show their effectiveness on the airfoil design problem. GPEME finds the best results, while ESAO converges the fastest on the initial stage. When the computing resource is more limited, ESAO is preferred.

V. Conclusion

In this study, we proposed an evolutionary sampling assisted optimization (ESAO) method for high dimensional expensive problems. ESAO considers global exploration and local exploitation simultaneously. The main work can be concluded as follows,

(1) A global RBF model is built with all existing samples to find the most promising offspring generated using mutation and crossover operators. A better sample will replace its parent's position. A local model is also built with a certain number of current best points. The global and local searches alternate when no better solution is found.

(2) The idea of adding better solution from the local search into the population is effective for ESAO. It is found that the local search will have more NFE than that of the global search. Correspondingly, the local search finds more NTI than that of the global search. RBF is more suitable to be the local model than Kriging, since the results of ESAO-R are much better than those of ESAO-K.

(3) ESAO is compared with GPEME, SA-COSO and SHPSO on 22 test problems. Results show that ESAO finds better results than GPEME and SA-COSO on most of the test problems. Compared to SHPSO, the performance of ESAO on 100D problems should be improved. Another promising feature of ESAO is that ESAO converges faster than the GPEME and SA-COSO on most of the test problems, and is comparable with SHPSO. Finally, ESAO is applied to a 12 variable airfoil design problem to prove its effectiveness.

The evolutionary sampling method used in the proposed algorithm is not limited to DE. It provides a framework and sheds some lights on the behavior and dynamics of the algorithm elements for future development. Next research will test the evolutionary operators of PSO instead of DE.

Acknowledgment

Supports from National Natural Science Foundation of China (Grant No. 51375389, Grant No. 51875466, Grant No. 51805436) are gratefully acknowledged. The authors gratefully acknowledge financial support from China Scholarship Council. We thank Dr. Di Wu for helping the implementation of Xfoil.

References

[1] R. H. Myers, D. C. Montgomery, *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*, Wiley, New York, 1995.
[2] J. D. Martin, T. W. Simpson, "Use of kriging models to approximate deterministic computer models," *AIAA J.*, vol. 43, no. 4, pp. 853-863, 2005.
[3] N. Dyn, D. Levin, S. Rippa, "Numerical procedures for surface fitting of scattered data by radial functions," *SIAM J Sci. Statist. Comput.*, vol. 7, no. 2, pp. 639-659, 1986.
[4] S. R. Gunn, "Support vector machines for classification and regression," Tech. rep., *University* of Southampton, UK, 1997.
[5] J. H. Friedman, "Multivariate adaptive *regression* splines," *The Ann. Stat.*, pp. 1-67, 1991.
[6] R. Jin, W. Chen, T. W. *Simpson*, "Comparative studies of metamodelling techniques under multiple modelling criteria," *Struct. Multidiscip. Optim.*, vol. 23, no. 1, pp. 1-13, 2001.
[7] A. I. J. Forrester, A. J. Keane, "*Recent* advances in surrogate-based optimization," *Prog. Aerosp. Sci.*, vol. 45, no. (1-3), pp. 50-79, 2009.
[8] D. R. Jones, M. Schonlau, W. J. *Welch*, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, pp. 455-492, 1998.
[9] D. R. Jones, "A *taxonomy* of global optimization methods based on response surfaces," *J. Glob. Optim.*, vol. 21, no. 4, pp. 345-383, 2001.
[10] J. Gu, G. Y. Li, Z. Dong, "Hybrid and adaptive meta-model-based global *optimization*," *Eng. Optim.*, vol. 44, no. 1, pp. 87-104, 2012.
[11] H. Dong, B. Song, Z. Dong, P. Wang, "Multi-start Space Reduction (MSSR) surrogate-based global optimization method," *Struct. Multidiscip. Optim.*, vol. 54, no. 4, pp. 907-926, 2016.
[12] P. Ye, G. Pan, "Global optimization method using ensemble of metamodels based on fuzzy clustering for design space reduction," *Eng. Comput.*, vol. 33, no. 3, pp. 573-585, 2017.
[13] R. Shi, L. Liu, T. Long, J. Liu, "Sequential radial basis function using support vector machine for expensive design optimization," *AIAA J.*, vol. 55, no. 1, pp. 214-227, 2017.
[14] D. Huachao, S. Baowei, W. PENG, "Hybrid surrogate-based optimization using space reduction (HSOSR) for expensive black-box functions." *Applied Soft Computing*, vol. 64, pp. 641-655, 2018.
[15] D. Huachao, S. Baowei, W. PENG, "SCGOSR: Surrogate-based constrained global optimization using space reduction." *Applied Soft Computing*, vol. 65, pp. 462-477, 2018.
[16] D. Ginsbourger, R. Le Riche, L. Carraro, "Kriging is well-suited to parallelize optimization," *Computational Intelligence in Expensive Optimization Problems*. Springer, Berlin, Heidelberg, 2010, pp. 131-162.
[17] F. A. C. Viana, R, T. Haftka, L. T. Watson, "Efficient global optimization algorithm assisted by multiple surrogate techniques," *J. Glob. Optim.*, vol. 56, no. 2, pp. 669-689, 2013.
[18] J. Liu, W. P. Song, Z. H. Han, Y. Zhang, "Efficient aerodynamic shape optimization of transonic wings using a parallel infilling strategy and surrogate models," *Struct. Multidiscip. Optim.*, vol. 55, no. 3, pp. 925-943, 2013.
[19] R. T. Haftka, D. Villanueva, A. Chaudhuri, "Parallel surrogate-assisted global optimization with expensive functions–a survey," *Struct. Multidiscip. Optim.*, vol. 54, no. 1, pp. 3-13, 2016.
[20] R. G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *J. Comput. Sci.*, vol. 5, no. 1, pp. 12–23, 2014.
[21] S. Shan, G. G. Wang, "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Struct. Multidiscip. Optim.*, vol. 41, no. 2, pp. 219-241, 2010.
[22] S. Shan, G. G. Wang, "Metamodeling for high dimensional simulation-based design problems," *J. Mech. Des.*, vol. 132, no. 5, pp. 051009, 2010.
[23] D. Wu, K. H. Hajikolaei, G. G. Wang, "Employing partial metamodels for optimization with scarce samples," *Struct. Multidiscip. Optim.*, no. 1, pp. 1-15, 2017.
[24] D. Wu, E. Coatanea, G. G. Wang, "Dimension Reduction and Decomposition Using Causal Graph and Qualitative Analysis for Aircraft Concept Design Optimization," *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, AEME, 2017.
[25] K. H. Hajikolaei, G. H. Cheng, G. G. Wang, "Optimization on metamodeling-supported iterative decomposition," *J. Mech. Des.*, vol. 138, no. 2, pp. 021401, 2015.
[26] K. C. Giannakoglou, Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, *38*(1), 43-76, 2002.
[27] D. Buche, N. N. Schraudolph, P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Transactions on Systems, Man, and Cybernetics*, Part C , vol. 35, no. 2, pp. 183-194, 2005.
[28] M. T. M. Emmerich, K. C. Giannakoglou, B. Naujoks, "Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 421-439, 2006.
[29] Y. S. Ong, P. B. Nair, A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA journal*, vol. 41, no. 4, pp. 687-696, 2003.
[30] Y. Tang, J. Chen, J. Wei, "A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions," *Eng. Optim.*, vol. 45, no,5, pp. 557-576,2013.

[31] L. Vincenzi, M. Savoia, "Coupling response surface and differential evolution for parameter identification problems," *Computer‐Aided Civil and Infrastructure Engineering*, vol, 30, no. 5, pp. 376-393, 2015.

[32] L. Vincenzi, P. Gambarelli, "A proper infill sampling strategy for improving the speed performance of a Surrogate-Assisted Evolutionary Algorithm," *Computers & Structures*, vol, 178, pp. 58-70, 2017.

[33] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, 2002.

[34] Z. Zhou, Y. Ong, P. Nair, A. Keane, and K. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Systems, Man, Cybern., C: Appl. Rev.*, vol. 37, no. 1, pp.66–76, 2007.

[35] D. Lim, Y. Jin, Y. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evolut. Comput.*, vol. 14, no. 3, pp. 329–355, 2010.

[36] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61-70, 2011.

[37] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 326–347, Jun. 2014.

[38] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[39] H. Wang, Y. Jin, J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2664-2677, 2017.

[40] C. Sun, Y. Jin, R. Cheng, et al, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644-660, 2017.

[41] H Yu, Y Tan, J Zeng, C Sun, Y Jin, "Surrogate-assisted hierarchical particle swarm optimization," Information Sciences, vol. 454, pp. 59-72, 2018.

[42] S. Das, P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4-31, Apr. 2011.

[43] T. Goel, R. T. Haftka, W. Shyy, N. V. Queipo, "Ensemble of surrogates," *Struct. Multidiscip. Optim.*, vol. 33, no. 3, pp. 199-216, 2007.

[44] E. Acar, M. Rais-Rohani, "Ensemble of metamodels with optimized weight factors," *Struct. Multidiscip. Optim.*, vol. 37, no. 3, pp. 279-294, 2009.

[45] R. Jin, W. Chen, A. Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments," *Journal of Statistical Planning and Inference*, Vol. 134, pp. 268–287, 2005.

[46] Y Wang, Q Wu, A P Punnen, et al, "Adaptive tabu search with strategic oscillation for the bipartite boolean quadratic programming problem with partitioned variables," Information Sciences, vol. 450, pp. 284-300, 2018.

[47] FAC Viana, "SURROGATES Toolbox User's Guide," Version 3.0, 2011, available at http://sites.google.com/site/-felipeacviana/surrogatestoolbox.

[48] M. Locatelli, A Note on the Griewank Test Function, Journal of Global Optimization, vol. 25, no. 2, pp. 169–174, 2003.

[49] R Cheng, Y Jin, "A social learning particle swarm optimization algorithm for scalable optimization," Information Sciences, Vol. 291, pp. 43-60, 2015.

[50] XFOIL (2013) [Online]. Available at: http://web.mit.edu/drela/Public/ web/xfoil/.