2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Behavioral Study of the Surrogate Model-aware Evolutionary Search Framework

Bo Liu, Qin Chen, Qingfu Zhang, Georges Gielen and Vic Grout

Abstract— **The surrogate model-aware evolutionary search (SMAS) framework is an emerging model management method for surrogate model assisted evolutionary algorithms (SAEAs). SAEAs based on SMAS outperform several state-of-the-art SAEAs using other model management methods and show promising results in real-world computationally expensive optimization problems. However, there is little behavioral study of the SMAS framework, and appropriate rules for its search strategy, training data selection and key parameter selection for different types of problems have not been provided yet. In this paper, with a newly proposed training data selection method, the SMAS framework's behaviour with different search strategies and training data selection methods is investigated. The empirical rules in terms of problem characteristics are obtained and the method to construct an SAEA based on the SMAS framework is updated. Experiments using 24 widely used benchmark test problems and the test problems in the CEC 2014 competition of computationally expensive optimization are carried out, which validate the proposed empirical rules.**

I. Introduction

Many real-world optimization problems require computationally expensive fitness function evaluations [1], [2]. Directly applying evolutionary algorithms (EAs) is often not feasible because a large number of time-consuming function evaluations are unaffordable. Surrogate model assisted evolutionary algorithms (SAEAs) are a recent promising approach for dealing with such expensive optimization problems. In SAEA, a surrogate model is employed to replace computationally expensive exact function evaluations. Surrogate models are approximation models of the fitness function that are much cheaper to evaluate than the real evaluation and the additional surrogate modeling process is often not expensive. Due to this, the computational cost can be reduced significantly.

Surrogate model management technique [3] is one of the most important research topics in SAEA research. Model management investigates optimal ways to have surrogate modeling and evolutionary search collaborate, so as to make them working together harmoniously. For a successful SAEA, a good balance between the effectiveness and the efficiency of the optimization must be made, which are often contradictory. To obtain good optimization ability, a high quality surrogate model is necessary, but this often implies more training data points, which can only be obtained by using more expensive evaluations. At present, several successful model management methods are receiving much attention and many SAEAs are constructed based on them. They include the generation control (GC) framework [4], the trust-region local search (TLS) framework [5], [6], the prescreening-based framework [7] (some references use individual-based method) and surrogate model-aware evolutionary search (SMAS) framework [8].

Standard EA processes are adopted in the GC framework, the TLS framework and the prescreening-based framework. The GC framework and the TLS framework can often obtain good solution quality but need more expensive function evaluations, while the prescreening-based framework often shows high efficiency but the solution quality and robustness still need to be improved. SMAS possesses advantages with respect to both optimization quality and efficiency. The GPEME algorithm [8] based on SMAS and Gaussian process modeling obtains comparable results but uses 12% to 50% of the number of exact function evaluations compared to other frameworks based on more than ten benchmark problems [5], [6], [7]. Unlike other frameworks relying on standard EAs, SMAS develops a new evolutionary search scheme considering high-quality modeling by controlling the locations of candidate solutions in the optimization. SMAS always focuses the search in the current promising subregion and gradually moves this subregion for exploration. Because the training data points and the candidates to be predicted are often distributed in the same small subregion, rather than the whole decision space like standard EA, a high-quality surrogate model can often be constructed with much less training data points (exact evaluations).

However, owing to the new search framework of SMAS, the population diversity and exploration ability become new problems, which are often not concerns for SAEA frameworks using standard EAs. On the other hand, there are also research works stating that standard EAs may have too much randomness or excessive diversity [9]. Hence, it is interesting to investigate the behavior of the SMAS framework in terms of search ability and to provide empirical rules for key parameter and search strategy selection. A typical SAEA based on SMAS is GPEME [8], using differential evolution (DE) operators and Gaussian Process (GP) modeling. DE operators will be used here as the example to explore general search behaviors of SMAS. More specifically, this paper aims to answer the following questions:

- [8] shows good search ability of SMAS using only 1000

B. Liu and V. Grout are with the Department of Computing, Glyndwr University, Wrexham, U.K. (e-mail: {b.liu,v.grout}@glyndwr.ac.uk, liubo168@gmail.com).

Q. Chen is with China Aerodynamic Research & Development Center, China. (e-mail: chenqin@fdiicardc.org)

Q. Zhang is with City University of Hong Kong, Kowloon, Hong Kong SAR. (e-mail: qingfu.zhang@cityu.edu.hk)

G. Gielen is with ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium (e-mail: Georges.Gielen@esat.kuleuven.be).

exact evaluations for most test problems. Can the search still be effective when using more evaluations, or can the results hardly be improved even if more evaluations are allowed?
- When more evaluations are allowed, what are the search behaviors of SMAS for unimodal, multimodal and discrete expensive optimization problems?
- There is plenty of research on parameter setting strategies of DE. Are the empirical rules also applicable to SMAS-based SAEAs using DE operators?
- DE/best/1 strategy (giving often the fastest convergence but the least diversity) is used for the SMAS framework in both an algorithmic investigation [8] and practical applications [10]. Can this be justified, or is there any other feasible or even better DE search strategy for SMAS-based SAEAs?

Besides the number of training data points and their locations, selecting which training data points to use and the surrogate model style are main factors affecting the surrogate model quality in an SAEA. Using a set of top-ranking solutions [6], using a set of neighbouring solutions of a point to be predicted [7], [5] and using a set of latest evaluations [11], [8] are widely used modeling styles. Training a single surrogate model for a whole population of individuals is shown to be more effective than training a separate surrogate model for each individual [11]. GPEME uses the same method as in [11], which constructs a single surrogate model for all the candidates in each iteration with a set of latest evaluated data points. [12] improves the training data selection method of SMAS and shows promising result for mm-wave integrated circuit design optimization problem. This paper proposes yet another training data point selection method and compares it with the method in [12] in collaboration with different search strategies. For comparison purposes, GP modeling is used in all of the methods.

The remainder of this paper is organized as follows. Section II reviews the SMAS framework. GP modeling and the DE operators are also introduced briefly. Section III analyzes the behavior of SMAS considering key parameters, search strategies and the training data selection method for different kinds of problems. Experimental verifications are provided in Section IV. Section V provides the concluding remarks.

II. Review of the SMAS Framework

The SMAS framework can be described as follows:

Step 1: Sample α (often small) solutions from the decision space $[a,b]^d$, evaluate the real function values of all these solutions and let them form the initial database.

Step 2: If a preset stopping criterion is met, output the best solution from the database; otherwise go to step 3.

Step 3: Select the λ best solutions from the database to form a population P.

Step 4: Apply the evolutionary operators on P to generate λ child solutions.

Step 5: Select τ training data to construct a surrogate model and use it to prescreen the λ child solutions generated in Step 4.

Step 6: Evaluate the real function value of the estimated best child solution from Step 5. Add this evaluated solution and its function value to the database. Go back to Step 2.

In terms of surrogate modeling, more effective training data can be expected in SMAS than those generated by a standard EA. In each iteration, the λ current best candidate solutions form the parent population (it is reasonable to assume that the search focuses on the promising subregion) and the best candidate based on prescreening in the child population is selected to replace the worst one in the parent population. In this way, only at most one candidate is changed in the parent population in each iteration; so the best candidate in the child solutions in several consecutive iterations may be quite near to each other (they will then be evaluated and are used as training data points). Therefore, the training data points describing the current promising region can be much denser compared to those generated by a standard EA population updating mechanism, which may spread in different subregions of the decision space while there may not be sufficient training data points around the candidate solutions to be prescreened.

GPEME [8] is an SAEA based on the SMAS framework. It uses GP [13] for surrogate modeling and DE operators [14] for evolutionary search. GP modeling is widely used in SAEAs and more details can be found in [7], [15]. DE is an effective and popular global optimization algorithm for real parameter optimization. It uses a differential operator to create new candidate solutions [14]. Although there are quite a few different DE variants, only the DE/best/1 strategy was adopted in SAEAs based on the SMAS framework. DE mutation strategies trade off the convergence speed and population diversity in different manners, and the DE/best/1 strategy often has the highest convergence speed but the lowest population diversity. In this paper, the following three DE mutation strategies ((1) to (3)) are investigated:

(1)DE/best/1

$$v_i = x^{best} + F \cdot (x^{r_1} - x^{r_2}) \tag{1}$$

where x^{best} is the best individual in P (see Step 3) and x^{r_1} and x^{r_2} are two different solutions randomly selected from P and are also different from x^{best}. v_i is the i^{th} mutant vector in the population after mutation. $F \in (0,2]$ is a control parameter, often called the scaling factor [14].

(2)DE/rand/1

$$v_i = x^{r_3} + F \cdot (x^{r_1} - x^{r_2}) \tag{2}$$

Compared to DE/best/1, x^{best} is replaced by a randomly selected solution x^{r_3} that is also different from x^{r_1} and x^{r_2}.

(3)DE/current-to-best/1 [1]

$$v_i = x^i + F \cdot (x^{best} - x^i) + F \cdot (x^{r_1} - x^{r_2}) \tag{3}$$

[1]This mutation strategy is also referred to as DE/target-to-best/1.

where x^i is the i^{th} vector in the current population.

III. Behavioral Analysis of the SMAS Framework

A. Characteristics of function landscapes

Compared to conventional EA, SAEA has the effect of smoothing the multimodal landscapes of complex problems [16]. Good surrogate models are able to catch the important trend of a function landscape, which is beneficial for jumping out of local optima. Hence, we empirically divide function landscapes into three categories: (A) functions without very complex general trends, (B) functions with (very) rugged landscapes, and (C) functions with integer / discrete variables. Fig. 1 shows two typical function landscapes. It can be seen that although the 2-dimensional Ackley function (Fig. 1 (A)) is multimodal, its general trend is much simpler than that of the 2-dimensional Rastrigin function (Fig. 1 (B)). For functions with rugged landscapes, constructing a surrogate model which is effective for prediction / prescreening may become difficult. For discontinuous optimization problems, although a good surrogate model can facilitate the prediction / prescreening of promising candidate solutions, the promising solutions must first be generated, and the challenges for traditional EAs in terms of search ability remain for SAEA. As said above, the algorithm structure of SMAS is different from a standard EA. This motivates the investigation of the behavior of SMAS in the context of the above three problem categories.

B. Key parameter selection rules

DE search has been used in SAEAs based on SMAS [8] and is also the example method in this study. The three critical parameters are the scaling factor F, the crossover rate CR and the population size λ. In the past decades, various research has investigated the choice of F and CR and different kinds of methods have been proposed [14], [17]. In standard DE, F is suggested to be set around 0.5 or larger to balance the exploration and exploitation. In SMAS, a large F is often necessary. The reason is that SMAS always uses the λ best solutions from the database as the parent population, which emphasizes exploitation. To maintain the exploration ability, a large F is needed. Our pilot experiments have shown that $F \in [0.75, 0.95]$ often achieves good results. The crossover rate, CR, on the other hand, is problem specific. Good values of CR generally fall into a small range for a given problem [17]. This implies that a self-adaptation mechanism for CR in the SMAS framework is useful. On the other hand, fast convergence is needed for expensive optimization problems, and it is recognized that large CR values can speed up the convergence to a large extent [14]. Hence, another alternative is to use a relatively large CR, such as $CR \in [0.7, 0.9]$. For problems with a not very complex general trend (category A in Section III (A)), it is predicted that this is an appropriate choice. However, for category B problems, the search may be trapped into local optima. For simplicity, $F = 0.8$ and $CR = 0.8$ are used for all the test problems in this paper. Like standard DE, the population size relies on the complexity of the problem. An experimental study [8], [12] shows that $\lambda \in [30, 50]$ often works well for most problems with several tens of variables. On the other hand, for SMAS, using a larger λ can help for complex problems (e.g., category B problems) when a certain search strategy and training data selection method are used, as will be verified in Section IV. For those problems, using λ between $4 \times d$ to $6 \times d$ is recommended according to our pilot experiments, where d is the number of variables.

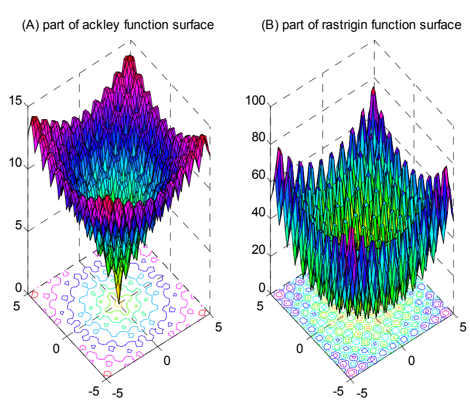

Fig. 1. Two typical function landscapes

C. Training data selection methods

The latest training data selection method for SMAS-based SAEA is in [12], which is as follows:

method for promising area-based training data selection (PAS)

1: Calculate the median of the λ child solutions to obtain the vector mx.

2: Take the nearest $c_1 \times d$ solutions to mx in the database (based on Euclidean distance).

The coefficient c_1 is often selected from $[4, 6]$ [12]. According to the general idea of SMAS, both the λ child solutions and the training data points are around the current promising area. Thus, a set of training data points in proportion to the number of variables are selected to model the general trend of the targeted area. Because the nearest neighbouring points are more useful than points far from the targeted area [7], [6], the training data points are sorted based on their distance to mx. On the other hand, for rugged landscapes (category B in Section III (A)), the surrogate model may not be sufficient to describe the complex shape of the current promising area. Therefore, an alternative method is proposed:

method for individual solution-based training data selection (ISS)

1: For each solution in the λ child solutions, take the nearest $c_2 \times d$ solutions in the database (based on Euclidean distance) as temporary training data points.

2: Combine all the temporary training data points and remove the duplicated ones.

To trade off the model quality and the training cost, empirical results suggest $c_2 \in [0.5, 1]$. This method emphasizes the modeling of the area surrounding each child solution and also builds a single surrogate model for the whole population to improve the model quality, instead of building a separate model for each child solution (see [11] for the reason).

D. DE search strategies

DE/best/1, DE/rand/1 and DE/current-to-best/1 trade off the convergence speed and the population diversity in different manners and are widely used in standard DE, especially the former two. The key of SMAS is to concentrate the search and the surrogate modeling in the current promising subregion, which is achieved by two factors: (1) the population update, (2) the mutation and crossover. Therefore, it is necessary to move the child population towards the current best solution in the DE mutation. The DE/current-to-best/1 and DE/best/1 strategies are thus appropriate to be used in SMAS, and the former can lead to a higher diversity. Although DE/rand/1 is widely used in standard DE, it is not an appropriate strategy for SMAS, because child solutions spreading in different subregions of the decision space may be generated and a high-quality surrogate model is often difficult to be constructed using such training data points. In Section IV, this conclusion will be verified with an example. Even assuming that an acceptable surrogate model is constructed, the convergence is quite slow using DE/rand/1.

Whether the additional population diversity of DE/current-to-best/1 compared to DE/best/1 offers substantial help or not is function landscape specific. For a category A landscape, where the general trend is not very complex, DE/best/1 should be able to obtain a reasonably good result because the probability is often low that the algorithm gets stuck in local optima when the general trend is correctly modeled. When using DE/current-to-best/1, the evolution speed may be slower but the search ability can be better. It is therefore difficult to compare these two strategies without experimental study. Nevertheless, a tendency is that DE/current-to-best/1 can have advantages when the number of variables increases. The two reasons are: (1) More exploration is needed for problems with a large scale; (2) When the number of variables is large, given a certain number of training data (limited by the allowed number of exact function evaluations), high quality surrogate modeling becomes more difficult in many cases, and some promising solutions might not correctly be prescreened. DE/current-to-best/1 provides more promising candidate solutions of different types because of the reasonable population diversity. This increases the probability that at least some type of promising solutions can be prescreened correctly by the available surrogate model. For a category B landscape (rugged) and a category C landscape (discontinuous), DE/current-to-best/1 has advantages because of the enhanced exploration ability. For a category C landscape, using the simulation method from [8], our pilot experiments show that a good result cannot be obtained using DE/best/1, even when the quality of the surrogate model is high. Note that for complex problems, the ISS method should be used together with DE/current-to-best/1. The PAS method has often difficulty to achieve high-quality surrogate modeling for category B and category C landscapes, the improved exploration ability of DE/current-to-best/1 therefore makes little sense without the support of good surrogate models. When using DE/best/1 together with the PAS method for category B and category C landscapes, the convergence can be quite fast and a reasonably acceptable solution may be obtained, because at least a portion of the general trend can often be modeled correctly. For some real-world expensive optimization problems, the computing budget is typically very tight [2] and obtaining highly optimized solutions is often not a requirement. In such circumstance, using DE/best/1 and the PAS method is also a good choice.

Based on the above analysis, the rules for selecting the training data and the search strategy are summarized as follows:

- For continuous optimization problems with a very tight computing budget, use the DE/best/1 mutation strategy with the PAS method.
- For discrete optimization problems, use the DE/current-to-best/1 mutation strategy with the ISS method.
- If there is prior knowledge of the problem landscape and the computing budget is not so tight, use the DE/current-to-best/1 mutation strategy and use the PAS method for category A landscapes and the ISS method for category B landscapes.
- If there is no prior knowledge of the problem landscape and the computing budget is not so tight, use the DE/current-to-best/1 mutation strategy with the ISS method.

IV. Experimental Study

A. Test problems and classifications

24 widely used benchmark test problems are used, which are shown in the Appendix. 20 runs are performed for each of them. For 10-dimensional problems, 1000 exact evaluations are used; for 20-dimensional problems, 1500 exact evaluations are used; for 30-dimensional problems, 2000 exact evaluations are used. The behaviors of GPEME [8] using different combinations of methods for training data and mutation strategy selection are studied. All the F and CR are set to 0.8. In the training data selection methods, c_1 is set to 6 and c_2 is set to 0.5 for all the problems. In the lower confidence bound prescreening, ω is set to 2. The number of initial samples is set to $5 \times d$ according to [12]. For 10-dimensional problems, λ is set to 30; for 20-dimensional

[2] The tightness of the computing budget is related to the problem complexity, but it may not be known beforehand. Hence, the evaluation time of a single solution versus the practical time to finish the optimization is considered here.

TABLE I

CLASSIFICATION OF THE TEST PROBLEMS

Category	Problems
A	Sphere (TF1-TF3), Ellipsoid (TF4-TF6), Rotated ellipsoid (TF7-TF9), Ackley (TF13-TF15), Griewank (TF16-TF18)
B	Rosenbrock (TF19-TF21), Rastrigin (TF22-TF24)
C	Step (TF10-TF12)

TABLE II

STATISTICS OF THE BEST FUNCTION VALUES OBTAINED FOR TF1-TF24 USING DE/BEST/1 WITH THE PAS METHOD

Problem	best	worst	average	std
TF1	1.5e-20	1.105e-18	2.71e-19	3.78e-19
TF2	3.67e-14	4.037e-13	1.839e-13	1.510e-13
TF3	2.09e-11	3.442e-10	1.300e-10	9.10e-11
TF4	0	1.054e-7	1.05e-8	3.33e-8
TF5	5.4e-13	3.892e-11	5.79e-12	1.178e-11
TF6	4.38e-10	7.097e-9	2.500e-9	2.234e-9
TF7	1.72e-16	4.462e-7	4.46e-8	1.411e-7
TF8	4.4e-11	6.217e-9	1.626e-9	1.967e-9
TF9	1.74e-6	8.298e-5	1.983e-5	2.860e-5
TF10	0	1.0000	0.3000	0.4830
TF11	0	5.0000	1.8000	1.3984
TF12	0	8.0000	3.2000	2.3476
TF13	0.0002	1.1551	0.1173	0.3647
TF14	0.0001	1.4235	0.2580	0.5472
TF15	0.0001	5.2219	1.4354	1.5602
TF16	0.0271	0.5063	0.2006	0.1673
TF17	0.0000	0.0172	0.0039	0.0066
TF18	0.0000	0.0123	0.0012	0.0039
TF19	0.7377	9.5938	3.0674	2.5592
TF20	4.0519	17.0733	13.8642	3.7562
TF21	22.5733	78.7295	31.1038	16.8010
TF22	5.9698	35.4970	18.6024	8.7344
TF23	15.0031	75.2502	37.8471	16.5290
TF24	40.7934	292.1517	116.9145	100.1807

problems, λ is set to 40 and for 30-dimensional problems, λ is set to 50.

The problem characteristics are analyzed and classified for verification purpose (see in Table I). Note that although the Ackley function and the more than 10-dimensional Griewank function are multimodal, their general trends are not very rugged, so they are classified to category A. For the Rosenbrock function, the general trend matches category A for some subregions, but near the narrow valley where the global optimum is located, the general trend is rugged, so it is classified into category B.

B. Performances and discussions

The statistics of the best function values obtained for TF1 to TF24 are reported in Table II, Table III and Table IV using different search strategy and training data selection methods.

The convergence trends (using the median value) are shown in Fig. 2 and Fig. 3.

Some observations can be derived from the above statistics. Firstly, for most of the problems with category A landscape, the convergence is clearly faster when using the PAS method than using the ISS method. On the other

TABLE III

STATISTICS OF THE BEST FUNCTION VALUES OBTAINED FOR TF1-TF24 USING DE/CURRENT-TO-BEST/1 WITH THE PAS METHOD

Problem	best	worst	average	std
TF1	3.5e-21	7.990e-19	1.136e-19	2.425e-19
TF2	6.4e-15	2.401e-13	5.26e-14	6.91e-14
TF3	4.5e-12	2.139e-10	5.02e-11	6.09e-11
TF4	1.5e-20	1.045e-18	4.01e-19	3.66e-19
TF5	6.18e-14	6.819e-13	3.373e-13	2.091e-13
TF6	7.5e-11	4.777e-9	1.004e-9	1.423e-9
TF7	1.06e-16	2.318e-14	3.10e-15	7.14e-15
TF8	8.8e-12	4.077e-10	8.17e-11	1.328e-10
TF9	1.3e-6	5.818e-4	9.31e-5	1.596e-4
TF10	0	1.0000	0.3000	0.4830
TF11	0	2.0000	1.1	0.6009
TF12	0	6.0000	3.0000	2.4037
TF13	0.0002	0.0014	0.0006	0.0004
TF14	0.0001	2.6997	0.2701	0.8537
TF15	0.0001	1.3404	0.5342	0.6356
TF16	0.0222	0.8789	0.1458	0.2771
TF17	0.0000	0.0123	0.0029	0.0051
TF18	0.0000	0.0465	0.0056	0.0139
TF19	1.6798	8.5125	4.2805	2.1279
TF20	14.6455	18.7358	15.8824	1.2498
TF21	25.3921	27.0233	26.5369	0.6615
TF22	3.9904	50.8884	24.8280	16.0412
TF23	19.9002	168.3490	84.2205	62.9253
TF24	35.8185	274.0313	225.7966	68.7886

TABLE IV

STATISTICS OF THE BEST FUNCTION VALUES OBTAINED FOR TF1-TF24 USING DE/CURRENT-TO-BEST/1 WITH THE ISS METHOD

Problem	best	worst	average	std
TF1	3.59e-14	2.439e-13	1.396e-13	6.31e-14
TF2	9.1e-8	2.015e-6	5.16e-7	5.75e-7
TF3	2.89e-5	2.236e-4	1.082e-4	6.07e-5
TF4	1.85e-14	3.768e-13	1.195e-13	1.108e-13
TF5	8.0e-9	6.903e-7	2.344e-7	2.423e-7
TF6	4.33e-5	4.419e-4	2.188e-4	1.514e-4
TF7	7.33e-12	3.361e-9	5.49e-10	1.023e-9
TF8	5.6e-6	3.470e-4	7.33e-5	1.170e-4
TF9	0.0230	0.2600	0.0783	0.0761
TF10	0	0	0	0
TF11	0	1	0.15	0.3663
TF12	0	2	0.8	0.8309
TF13	0.0123	0.0896	0.0496	0.0288
TF14	0.0034	2.3158	0.6101	0.8391
TF15	0.0814	1.3602	0.3339	0.3822
TF16	0.0172	0.0911	0.0411	0.0290
TF17	0.0003	0.0174	0.0075	0.0065
TF18	0.0135	0.4551	0.1414	0.1238
TF19	0.0778	4.0537	2.6176	1.1591
TF20	14.7134	19.0486	16.4620	1.4707
TF21	26.8365	29.4628	27.7370	0.9069
TF22	6.9647	16.9143	10.7456	2.8837
TF23	14.9263	84.3081	28.4888	20.2593
TF24	26.3768	196.2803	114.8459	59.3901

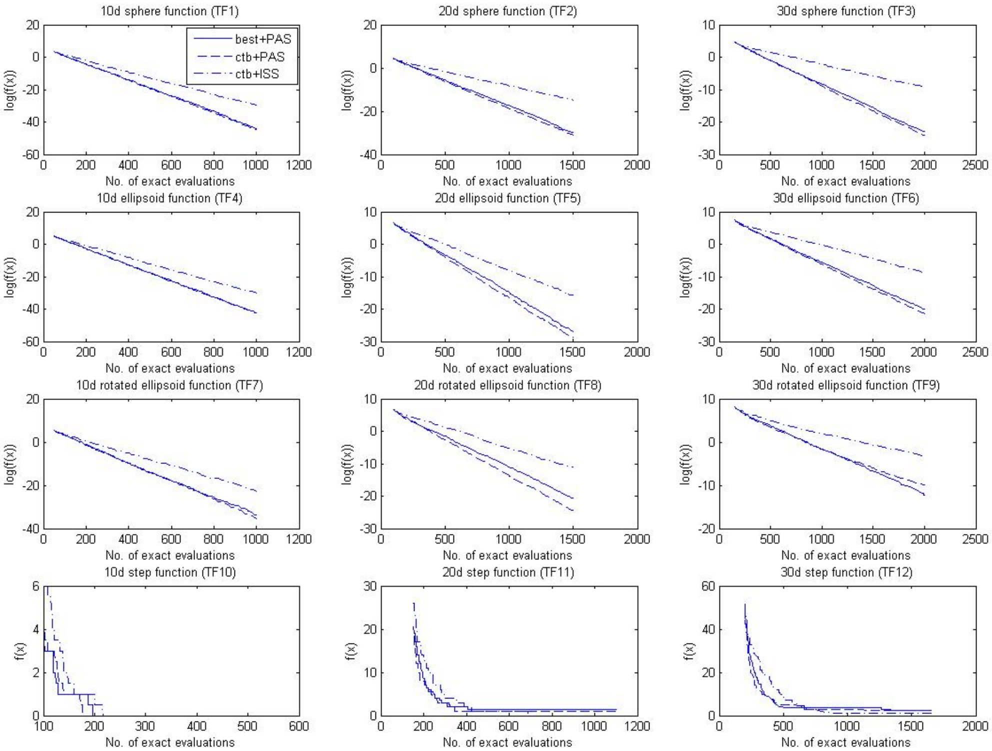

Fig. 2. Median of the best objective function values obtained for TF1 to TF12 over 20 runs

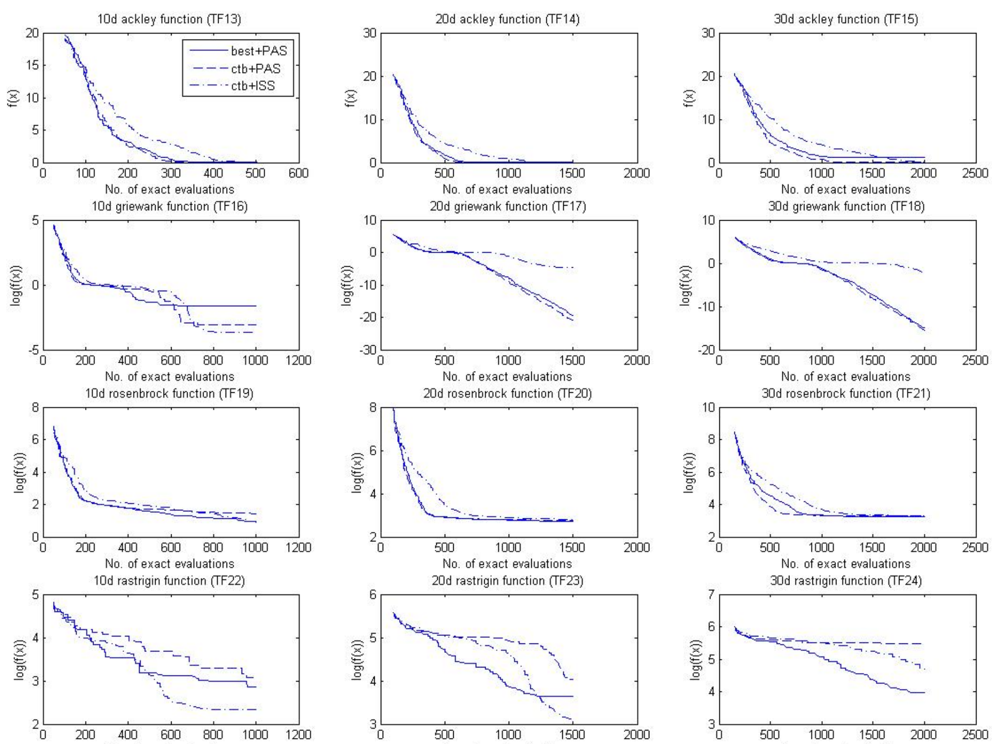

Fig. 3. Median of the best objective function values obtained for TF13 to TF24 over 20 runs

hand, both of them can obtain satisfactory results at last. When using the PAS method, the performance of using DE/current-to-best/1 is often slightly better than that of using the DE/best/1 strategy. For some 30-dimensional problems,

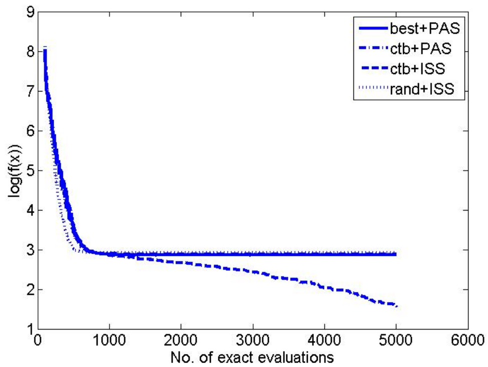

Fig. 4. Convergence curves for the 20-dimensional Rosenbrock function

DE/current-to-best/1 shows an obviously better performance because of the higher exploration ability, such as the 30-dimensional Ackley function. Secondly, for category B problems, it can be observed that although reasonably acceptable results are obtained for most problems, more evaluations are needed to have a clear picture on the convergence trend for several problems. For the 10-dimensional Rosenbrock function and the 10/20-dimensional Rastrigin function, the convergence trends are relatively clear. An observation is that DE/current-to-best/1 with the PAS method performs the worst, the reason of which has been described in Section III (D). Using DE/current-to-best/1 with the ISS method often obtains results with a better quality at last, while using DE/best/1 with the PAS method can obtain a faster convergence in the beginning and the final results are often reasonably acceptable. Thirdly, for category C landscapes, it can be seen that using DE/current-to-best/1 with the ISS method has obvious advantages compared to other methods. Fourthly, for continuous optimization, it can be seen that using DE/best/1 with the PAS method is often a good choice when the computing budget is very tight. These observations verify the analysis and the empirical rules presented in Section III.

Next, the experiment of using 5000 exact evaluations for the 20-dimensional Rosenbrock function is carried out (Fig. 4). The above conclusion is validated again. It can be observed that only using DE/current-to-best/1 with the ISS method can jump out of local optima. It also verifies that DE/rand/1 is not a feasible choice for SMAS. Although using DE/rand/1 has a higher population diversity than using DE/current-to-best/1, a high-quality surrogate model is difficult to be constructed by the generated training data points. Fig. 4 shows the convergence curves using $\lambda = 5 \times d$. Experiments show that when using $\lambda = 40$, local optima can still be jumped out with DE/current-to-best/1 and the ISS method, but the convergence speed is slower. This shows that with appropriate search and training data selection methods, enlarging the population size is an effective method to handle complex problems (e.g., category B landscapes) by the SMAS framework.

Test functions F1 to F24 from CEC 2014 competition [18] of computationally expensive optimization are used. For 10, 20 and 30-dimensional problems, the allowed number of exact function evaluations are 500, 1000 and 1500, respectively. According to the summarized rules in Section III, DE/best/1 with the PAS method is selected for such tight computing budget, except F10-F12 based on the Step function, where DE/current-to-best/1 with the ISS method is used and λ is set to $5 \times d$. The results are shown in Table V.

TABLE V

STATISTICS OF THE BEST FUNCTION VALUES OBTAINED FOR F1-F24 OVER 20 RUNS

Problem	best	worst	median	mean	std
F1	1.34E-08	2.74E-04	1.55E-07	1.41E-05	6.12E-05
F2	4.38E-08	2.64E-06	4.39E-07	7.08E-07	7.49E-07
F3	2.65E-07	6.29E-06	2.89E-06	2.96E-06	1.52E-06
F4	2.32E-08	2.38E-05	2.69E-07	2.56E-06	6.46E-06
F5	4.48E-07	1.80E-05	4.26E-06	1.42E-05	3.92E-05
F6	2.08E-05	6.17e-4	3.76E-05	7.33E-5	1.30E-4
F7	9.22E-07	4.62E-05	8.36E-06	1.28E-05	1.18E-05
F8	3.75E-05	6.24E-04	1.67E-04	1.99E-04	1.62E-04
F9	0.003	0.2924	0.0362	0.048	0.0645
F10	0	0	0	0	0
F11	0	1	0	0.1	0.3078
F12	0	2	1	0.6	0.5982
F13	3.15E-04	1.6463	8.95e-4	0.2238	0.5503
F14	5.25E-04	3.5719	0.0019	0.5792	1.1467
F15	0.0016	2.3169	0.9314	0.8854	0.8233
F16	0.0099	0.8459	0.0992	0.279	0.3346
F17	6.42E-06	0.057	0.0075	0.0083	0.0128
F18	4.21E-05	0.0148	2.90E-04	0.0027	0.0051
F19	0.2132	9.6114	5.288	5.2884	2.1684
F20	14.0813	81.6163	15.9776	22.1117	18.7612
F21	18.8617	164.15	33.5428	54.281	36.8855
F22	13.0691	85.4369	56.1722	50.5976	23.8103
F23	33.0937	167.0889	83.4525	88.7364	41.5934
F24	53.2462	310.5249	244.9737	198.7347	85.6271

V. CONCLUSIONS

The SMAS framework outperforms several other state-of-the-art surrogate management methods for computationally expensive optimization problems, but its search behavior has seldom been investigated. In this work, the parameter setting, the search strategy (trading off of convergence speed and population diversity), the training data selection method (further improving the surrogate model quality) and the correlation between these have been investigated for different types of function landscapes in the context of SMAS. Empirical rules for selecting the appropriate search strategy and the training data selection method have been proposed, which provide a reference for the appropriate use of SMAS. The effectiveness of the rules have been verified by experiments. Future work includes investigating the ensemble of different search and training data selection strategies and self-adaptive methods.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Junpeng Xiang, Department of Computing, Glyndwr University, for his efforts on experiment and statistics.

Appendix

Benchmark problems:

A. TF1, TF2, TF3: Sphere Problem

$$\begin{array}{l} min \quad f(x) = \sum_{i=1}^{d} x_i^2 \\ x \in [-5.12, 5.12], i = 1, \ldots, d \\ TF1 : d = 10, TF2 : d = 20, TF3 : d = 30 \end{array} \tag{4}$$

B. TF4, TF5, TF6: Ellipsoid Problem

$$\begin{array}{l} min \quad f(x) = \sum_{i=1}^{d} ix_i^2 \\ x \in [-5.12, 5.12], i = 1, \ldots, d \\ TF4 : d = 10, TF5 : d = 20, TF6 : d = 30 \end{array} \tag{5}$$

C. TF7, TF8, TF9: Rotated Ellipsoid Problem

$$\begin{array}{l} min \quad f(x) = \sum_{i=1}^{d} i \cdot xr_i^2 \\ xr = M * x, M \text{is from CEC 2014 competition [18]} \\ x \in [-5.12, 5.12], i = 1, \ldots, d \\ TF7 : d = 10, TF8 : d = 20, TF9 : d = 30 \end{array} \tag{6}$$

D. TF10, TF11, TF12: Step Problem

$$\begin{array}{l} min \quad f(x) = \sum_{i=1}^{d} (\lfloor x_i + 0.5 \rfloor)^2 \\ x \in [-5.12, 5.12], i = 1, \ldots, d \\ TF10 : d = 10, TF11 : d = 20, TF12 : d = 30 \end{array} \tag{7}$$

E. F13, F14, F15: Ackley Problem

$$\begin{array}{l} min \quad f(x) = -20e^{-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)} \\ x \in [-32.768, 32.768], i = 1, \ldots, d \\ TF13 : d = 10, TF14 : d = 20, TF15 : d = 30 \end{array} \tag{8}$$

F. TF16, TF17, TF18: Griewank Problem

$$\begin{array}{l} min \quad f(x) = 1 + \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} cos(\frac{x_i}{\sqrt{i}}) \\ x \in [-600, 600], i = 1, \ldots, d \\ TF16 : d = 10, TF17 : d = 20, TF18 : d = 30 \end{array} \tag{9}$$

G. TF19, TF20, TF21: Rosenbrock Problem

$$\begin{array}{l} min \quad f(x) = \sum_{i=1}^{d} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \\ x \in [-2.048, 2.048], i = 1, \ldots, d \\ TF19 : d = 10, TF20 : d = 20, TF21 : d = 30 \end{array} \tag{10}$$

H. TF22, TF23, TF24: Rastrigin Problem

$$\begin{array}{l} min \quad f(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10cos(2\pi x_i)) \\ x \in [-5.12, 5.12], i = 1, \ldots, d \\ TF22 : d = 10, TF23 : d = 20, TF24 : d = 30 \end{array} \tag{11}$$

References

[1] K. Giannakoglou, "Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence," *Progress in Aerospace Sciences*, vol. 38, no. 1, pp. 43–76, 2002.

[2] B. Liu, G. Gielen, and F. Fernández, "Automated design of analog and high-frequency circuits," *Studies in Computational Intelligence*, vol. 501, 2013.

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

[4] ——, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 9, no. 1, pp. 3–12, 2005.

[5] D. Lim, Y. Jin, Y. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[6] Z. Zhou, Y. Ong, P. Nair, A. Keane, and K. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 1, pp. 66–76, 2007.

[7] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

[8] B. Liu, Q. Zhang, and G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive black box optimization problems," *IEEE Transactions on Evolutionary Computation (To be published)*, 2013.

[9] K. Rasheed and H. Hirsh, "Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models." in *GECCO*, 2000, pp. 628–635.

[10] B. Liu, H. Aliakbarian, Z. Ma, G. Vandenbosch, G. Gielen, and P. Excell, "An efficient method for antenna design optimization based on evolutionary computation and machine learning techniques," *IEEE Transactions on Antennas & Propagation (To be published)*, 2014.

[11] R. Li, M. T. Emmerich, J. Eggermont, E. G. Bovenkamp, T. Back, J. Dijkstra, and J. H. Reiber, "Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, 2008, pp. 2764–2771.

[12] B. Liu, D. Zhao, and G. Gielen, "Gaspad: A general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (To be published)*, 2014.

[13] C. Rasmussen, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71, 2004.

[14] K. Price, R. Storn, and J. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer-Verlag New York Inc, 2005.

[15] B. Liu, Q. Zhang, F. V. Fernandez, and G. Gielen, "Self-adaptive lower confidence bound: A new general and effective prescreening method for gaussian process surrogate model assisted evolutionary algorithms," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–6.

[16] Y.-S. Ong, Z. Zhou, and D. Lim, "Curse and blessing of uncertainty in evolutionary algorithm using approximation," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 2928–2935.

[17] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 398–417, 2009.

[18] B. Liu, Q. Chen, Q. Zhang, J. J. Liang, P. N. Suganthan, and B. Qu, "Problem definitions and evaluation criteria for computationally expensive single objective numerical optimization," *Techical. Rep of Zhengzhou University, China and Nanyang Technological University, Singapore*, 2014.