Surrogate-Assisted Memetic Algorithm with Adaptive Patience Criterion for Computationally Expensive Optimization

Yunwei Zhang
Shaanxi Aerospace Flight Vehicle Design Key Laboratory, School of Astronautics
Northwestern Polytechnical University
Xi'an 710072, China
ywzhang@mail.nwpu.edu.cn

Chunlin Gong
Shaanxi Aerospace Flight Vehicle Design Key Laboratory, School of Astronautics
Northwestern Polytechnical University
Xi'an 710072, China
leonwood@nwpu.edu.cn

Chunna Li
Shaanxi Aerospace Flight Vehicle Design Key Laboratory, School of Astronautics
Northwestern Polytechnical University
Xi'an 710072, China
chunnali@nwpu.edu.cn

Abstract—**Surrogate-assisted memetic algorithm (SAMA) has been recognized to be an effective tool for computationally expensive optimization. The termination criterion of the local search in SAMA determines the allocation of limited computational resources between global and local search, and has a tremendous impact on the optimization performance. The commonly used termination criterion based on setting a limit to the local search depth can lead to premature termination or excessive stagnation iterations of the local search. This paper proposes a SAMA with adaptive patience criterion (SAMA/APC) to improve the efficiency of traditional SAMA. The SAMA/APC consists of three main subprocedures, which are carried out iteratively. First, the operators of differential evolution (DE) are employed for global exploration. Then, the proposed Kriging-based patience allocation strategy (KPAS) is performed, which adaptively allocates a *patience* value to each individual of the population according to two basic principles. Third, the trust-region search (TRS) is carried out on each individual for local exploitation. The TRS is a process of consuming the patience, and it terminates when the patience value is reduced to zero. The local optimum obtained by the TRS is returned back to the population of DE in the spirit of Lamarckian learning. Experimental studies on the CEC' 14 expensive optimization test suite demonstrate the efficiency of the proposed SAMA/APC.**

Keywords—Surrogate-assisted memetic algorithm, differential evolution, trust-region search, adaptive patience criterion, Kriging model

I. Introduction

Solving the computationally expensive optimization problems is a fundamental task in practical engineering applications, which, however, is also extremely challenging, since usually only a limited number of function evaluations (FEs) are affordable. Evolutionary algorithms (EAs) have achieved significant progress in applying to different fields [1]–[4]. However, they generally require excessive FEs for convergence and therefore cannot be directly utilized in computationally expensive optimization.

There are mainly two reasons for the poor convergence speed of the canonical EAs: 1) the information gained during optimization, i.e., the samples obtained by solving the time-consuming fitness functions, is wasted, which can actually be used to guide and accelerate the evolution; 2) the typical variation operators of EA, like crossover and mutation, are in nature good at exploring the whole search space but slow in exploiting a local region. Surrogate model and local search methods can be incorporated with EAs to address the above two issues, respectively. The resulting hybrid method, usually termed surrogate-assisted memetic algorithm (SAMA) [5], [6], can effectively improve the optimization efficiency.

SAMA has attracted increasing attention in recent years. Ong et al. [7] presented a parallel EA coupled with local radial basis function (RBF) network-assisted feasible sequential quadratic programming solver in the spirit of Lamarckian learning. Zhou et al. [8] proposed a novel SAMA framework, in which a global Kriging model is used to select promising individuals, and several local RBF networks are used to assist a trust-region search strategy. A surrogate-assisted cooperative swarm optimization algorithm is proposed in [9], in which two different forms of PSO optimizers focus respectively on global exploration and local exploitation.

Despite the extensive research on SAMA, few papers have been published investigating the termination criterion of the local search in SAMA, which determines the amount of computational resources assigned to the local search. When the total computational budget is limited, the termination criterion of local search controls the allocation of the computational resources between global and local search, hence affecting the balance between global exploration and local exploitation.

The commonly used termination criterion of local search in SAMA is to set a limit to the number of iterations of the local search, i.e., the local search depth (LSD) [10]. Sudholt [11] presented a rigorous theoretical analysis of the

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

TABLE I
DIFFERENT METHODS FOR DETERMINING THE LSD.

Method for determining LSD		Ref.
Fixed LSD		[6]–[8], [12]
Variable LSD	Increasing LSD	[13]
	Adaptive LSD	[14], [15], [16]

parameterization in memetic algorithms and stressed that the choice of the LSD has a tremendous impact on the optimization performance.

Methods in the literature for determining the LSD of SAMA can be divided into two categories: fixed LSD and variable LSD, as listed in Table I. In the work of [6]–[8], [12], the LSD is simply treated as a constant during the entire optimization, which is usually set to a relatively large value to achieve high local-search accuracy. However, it is often difficult to pre-specify the depth manually without prior knowledge. Moreover, a fixed LSD will lead to a fixed allocation ratio of computational resources between global and local search, which cannot be universally suitable for different search stages and problems with various degrees of nonlinearity. Variable LSD methods, including the increasing and adaptive LSD strategies, have been proposed to alleviate the above problems. Bambha et al. [13] compared the LSD to "temperature" and proposed the so-called simulated heating algorithm, which starts with a small LSD to focus on global search at the beginning and then gradually increases the LSD to put more effort on local exploitation and accelerate the convergence. Liu and Li [14] proposed a memetic algorithm with adaptive LSD, in which the LSD is adjusted dynamically during the optimization process according to the comparison of Average Fitness Increment (AFI) between local and global search. Molina et al. [15] divided all individuals of the population into three categories according to their fitness values, and assigned different LSDs to individuals in different categories. In the field of multi-objective optimization, Xu et al. [16] designed a Pareto-based variable depth search method, in which the search depth is dynamically adjusted during the search process to strike a balance between exploration and exploitation.

However, these termination criteria based on the LSD, regardless of the fixed or the variable LSD methods, immediately terminate the local search as long as its number of iterations reaches the maximum allowable LSD without considering the state of the local search, which can lead to two unexpected results:

1) The local search may be in a state of rapid decline, and better solutions are continuously being found. The sudden termination can cause it to lose the opportunity to find better or even the global optimal solution;
2) On the contrary, the local search procedure might also be trapped in a local optimum, and excessive stagnation iterations will waste the computational resources.

In this paper, we propose a SAMA with adaptive patience criterion (SAMA/APC) to address the abovementioned issues. The proposed SAMA/APC consists of three main subprocedures, which are carried out sequentially in a loop. First, the basic operators of differential evolution (DE) is performed for global exploration. Then, the proposed Kriging-based patience allocation strategy (KPAS) is carried out, which can allocate a *patience* value to each individual of the population of DE. Finally, the patience values of the individuals are utilized in the third subprocedure. The trust region search (TRS) is carried out on each individual for local exploitation, which is a patience-consuming process. The patience value is updated in each iteration of the TRS, and when it is reduced to zero, the TRS will be terminated. This termination criterion takes into account the state of the local search before terminating it so as to reduce the computational cost.

The rest of this paper is organized as follows. Section II describes the proposed SAMA/APC, and introduces the KPAS in detail, which is the main contribution of this paper. Experimental studies of the proposed method are presented in Section III. Finally, conclusions are summarized in Section IV.

II. SURROGATE-ASSISTED MEMETIC ALGORITHM WITH ADAPTIVE PATIENCE CRITERION

The three main subprocedures of the SAMA/APC are described in the following three subsections in detail, respectively, and the last subsection summarizes the complete procedure of the proposed SAMA/APC.

A. Differential evolution

The DE [17] is a population-based metaheuristic algorithm like other EAs. Denote by $\mathbf{P}^{(i)}=\{\mathbf{x}_1^{(i)},\mathbf{x}_2^{(i)},...,\mathbf{x}_{N_\mathrm{P}}^{(i)}\}$ the population of the ith generation, where N_P is the population size. First, the initial population $\mathbf{P}^{(0)}$ is generated by a space-filling design of experiment method. Then, the following three operators are used to generate the next population $\mathbf{P}^{(i+1)}$ based on $\mathbf{P}^{(i)}$.

First, the mutation operator [18] generates a mutant vector $\mathbf{v}_j^{(i)}$ for each individual $\mathbf{x}_j^{(i)}$ in $\mathbf{P}^{(i)}$:

$$\mathbf{v}_j^{(i)}=\mathbf{x}_{r_1}^{(i)}+F\bullet(\mathbf{x}_{r_2}^{(i)}-\mathbf{x}_{r_3}^{(i)}) \tag{1}$$

where r_1, r_2, and r_3 are three mutually different random integers selected from the set $\{1,...,j-1,j+1,...,N_\mathrm{P}\}$, and F is a positive mutation scaling factor.

Second, the crossover operator is applied to $\mathbf{v}_j^{(i)}$ and $\mathbf{x}_j^{(i)}$ to generate a trial vector $\mathbf{u}_j^{(i)}$:

$$u_{j,k}^{(i)}=\begin{cases} v_{j,k}^{(i)} & \text{if } \mathrm{rand}_j(0,1)\le \mathrm{CR} \text{ or } j{=}j_\mathrm{rand} \\ x_{j,k}^{(i)} & \text{otherwise} \end{cases} \tag{2}$$

where $u_{j,k}^{(i)}$, $v_{j,k}^{(i)}$, and $x_{j,k}^{(i)}$ are the kth $(k=1,2,...,D)$ component of $\mathbf{u}_j^{(i)}$, $\mathbf{v}_j^{(i)}$ and $\mathbf{x}_j^{(i)}$, respectively. D is the dimension of the search space, $\mathrm{rand}_j(0,1)$ is a random number generated from [0,1], $CR\in[0,1]$ is the crossover control parameter, and j_rand is a random integer chosen from $\{1,2,...,D\}$.

Finally, the selection operator determines whether or not $\mathbf{u}_j^{(i)}$ can replace $\mathbf{x}_j^{(i)}$ in the next generation $\mathbf{P}^{(i+1)}$:

$$\mathbf{x}_j^{(i+1)} = \begin{cases} \mathbf{u}_j^{(i)} & \text{if } f(\mathbf{u}_j^{(i)}) < f(\mathbf{x}_j^{(i)}) \\ \mathbf{x}_j^{(i)} & \text{otherwise} \end{cases} \tag{3}$$

where f is the expensive fitness function. It should be noted that all samples obtained by evaluating the fitness function are archived into a global database for constructing the Kriging model in the next subsection.

The inherent randomness of the mutation and crossover operators brings the global-exploration capability to DE, and the selection operator can effectively guide the population to the promising regions of the search space.

B. *Kriging-based patience allocation strategy*

This subsection introduces the proposed KPAS, which can adaptively assign a patience value to each individual of the population. The patience values are employed in the termination criterion of the local search in SAMA/APC, which is described in the following subsection.

Two basic principles are taken into account when allocating patience to the individuals:

1) From the *population* level, the total patience of all individuals in the population should gradually increase, similar to the idea of [13]. In the early stage, a relatively small patience value is adopted to focus on global search. As the optimization proceeds, the start points are approaching the global optimum, and a larger patience value can make computational resources gradually shifts from global search to local search so as to accelerate the convergence.
2) From the *individual* level, different individuals have different potentialities to find the global optimum through local search. Those individuals with small fitness values (for minimization problems) or large uncertainties are more likely to lead to better solutions. Therefore, a relatively larger patience value should be allocated to these more promising individuals.

The Kriging model [19], [20] is employed to determine the potentiality of each individual, because it can output not only the predictions, but also the standard deviation of the predictions [19]. This property can be utilized to calculate the probability of each individual to improve the current best solution.

The proposed KPAS consists of four subprocedures.

First, a Kriging model is constructed or updated for the objective function using all samples in the global database. One can refer to [19] and [20] for the detailed formulation and construction method of Kriging model.

Second, the Probability of Improvement (PoI) [8], [21] of each individual $\mathbf{x}_j^{(i)}$ in $\mathbf{P}^{(i)}$ can be calculated by:

$$\mathrm{PoI}(\mathbf{x}_j^{(i)}) = \Phi\left(\frac{f^* - \hat{f}(\mathbf{x}_j^{(i)})}{\sigma(\mathbf{x}_j^{(i)})}\right) \tag{4}$$

where $\hat{f}(\mathbf{x}_j^{(i)})$ are the predicted fitness of the Kriging model for $\mathbf{x}_j^{(i)}$, and $\sigma(\mathbf{x}_j^{(i)})$ is the standard deviation of the prediction. f^* represents the current best fitness in the global database. $\Phi(\bullet)$ is the normal cumulative distribution function. The POI of $\mathbf{x}_j^{(i)}$ indicates the probability of $\mathbf{x}_j^{(i)}$ to improve the current best solution.

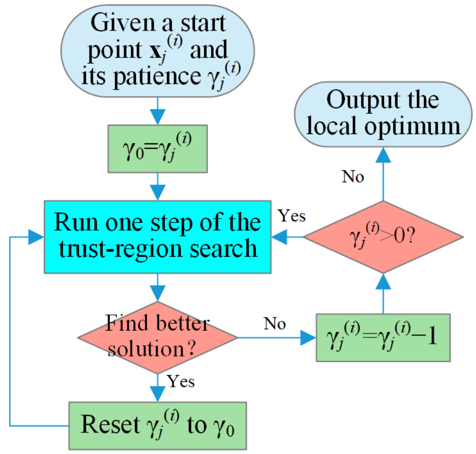

Fig. 1. Flowchart of the trust region search in the proposed SAMA/APC.

Third, the PoI values of all individuals in $\mathbf{P}^{(i)}$ are scaled to make their sum equal to one. The scaled PoI (SPOI) of $\mathbf{x}_j^{(i)}$ is calculated by

$$\mathrm{SPoI}(\mathbf{x}_j^{(i)}) = \frac{\mathrm{PoI}(\mathbf{x}_j^{(i)})}{\sum_{q=1}^{N_\mathrm{P}} \mathrm{PoI}(\mathbf{x}_q^{(i)})} \tag{5}$$

Finally, the patience value $\gamma_j^{(i)}$ allocated to $\mathbf{x}_j^{(i)}$ is calculated by

$$\gamma_j^{(i)} = i * N_\mathrm{P} * \mathrm{SPoI}(\mathbf{x}_j^{(i)}) \tag{6}$$

Since $\sum_{q=1}^{N_\mathrm{P}} \mathrm{PoI}(\mathbf{x}_q^{(i)}) = 1$, the total patience of all individuals is expressed as

$$\sum_{j=1}^{N_\mathrm{P}} \gamma_j^{(i)} = i * N_\mathrm{P} * \sum_{j=1}^{N_\mathrm{P}} \mathrm{SPoI}(\mathbf{x}_j^{(i)}) = i * N_\mathrm{P} \tag{7}$$

Therefore, on one hand, as the iteration counter i grows, the total patience of all individuals in $\mathbf{P}^{(i)}$ increases accordingly, which satisfies the first principle mentioned above. On the other hand, $\mathrm{SPoI}(\mathbf{x}_j^{(i)})$ can be considered as the share of $\gamma_j^{(i)}$ in the total patience of the current population, and the more promising individual, indicated by $\mathrm{SPoI}(\mathbf{x}_j^{(i)})$, has larger patience value, which satisfies the second principle.

In addition, a boundary constraint is set on $\gamma_j^{(i)}$. The lower bound is one so as to let all individuals have an opportunity to serve as the start points of local searches and make the start points more diverse. The upper bound is set to 10 to avoid wasting computational resources due to too large patience value.

C. *Trust region search with patience-based termination criterion*

The KPAS outputs all individuals of the population and their corresponding patience values to the TRS [22], [23] solver. Each individual is locally refined through the TRS.

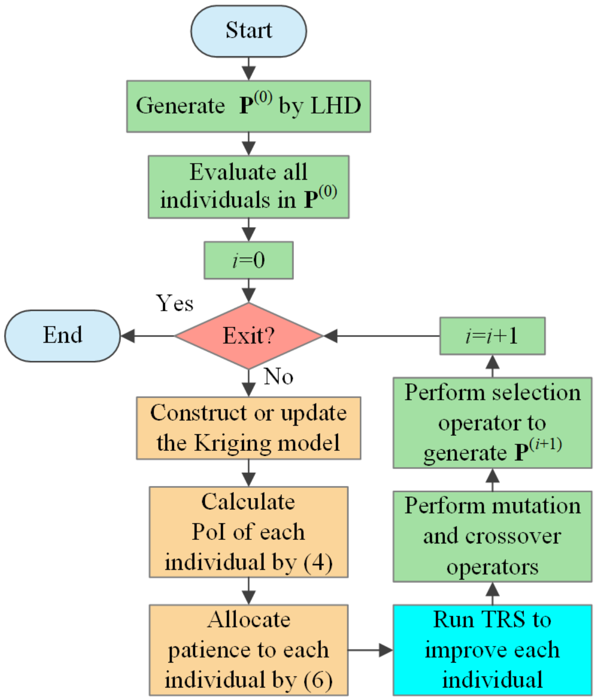

Fig. 2. Flowchart of the proposed SAMA/APC.

The TRS itself is an iterative process of solving a sequence of optimization subproblems with trust region constraints. This paper focuses on the application of the TRS. One can refer to [22] and [23] for the detailed formulation of the TRS, which is beyond the scope of this paper. Considering the disadvantages of the LSD-based termination criterion mentioned in Section I, we employ the patience-based criterion to determine when to terminate the TRS.

Fig. 1 presents the flowchart of the TRS in the proposed SAMA/APC. Given a start point $\mathbf{x}_j^{(i)}$ and its patience $\gamma_j^{(i)}$, the TRS can be seen as a patience-consuming process. First, the patience $\gamma_j^{(i)}$ of $\mathbf{x}_j^{(i)}$ is stored in a variable γ_0. Then, the iterative procedure of the TRS is carried out. After each step of the TRS, the patience $\gamma_j^{(i)}$ is updated as follows: if the current best solution stays unchanged in the last step, the patience $\gamma_j^{(i)}$ is decreased by one; if a better solution is found, $\gamma_j^{(i)}$ is reset to γ_0. When $\gamma_j^{(i)}$ is reduced to zero, the TRS will be terminated. Therefore, the initial patience $\gamma_j^{(i)}$ of $\mathbf{x}_j^{(i)}$ assigned by the proposed KPAS indirectly controls the LSD of the TRS and the amount of computational resources allocated for local exploitation. This patience-based termination criterion takes into account the state of the local search before terminating it so as to avoid both premature termination and excessive stagnation.

Finally, the obtained local optimum is then returned back to the population to replace the original individual in the spirit of Lamarckian learning [24].

TABLE II
SUMMARY OF THE BENCHMARK FUNCTIONS

No.	Functions	Dimension	Search space
13-15	Shifted Ackley's function	10,20,30	[-32, 32]
16-18	Shifted Griewank's function	10,20,30	[-600, 600]
19-21	Shifted Rotated Rosenbrock's function	10,20,30	[-20,20]
22-24	Shifted Rotated Rastrigin's function	10,20,30	[-20,20]

D. Complete procedure

The schematic diagram of the proposed SAMA/APC is shown in Fig. 2, and the complete procedure of the proposed SAMA/APC is summarized as follows:

Pseudocode of the proposed SAMA/APC

```
Set the global iteration counter i=0;
While the computational budget is not exhausted
    // global exploration
    If i==0
        Generate the initial population P^(0) by the LHD, and evaluate the individuals in P^(0) with the real objective functions[1];
    Else
        Run DE operators to generate new population P^(i) based on the last population P^(i-1);
    End if
    // Kriging-based patience allocation strategy
    Construct or update the Kriging model using all samples in the global database;
    Calculate the PoI of each individual in P^(i) using (4);
    Allocate the patience to each individual in P^(i) according to (6);
    // local search
    For each individual x_k (k = 1,2,...,N_P) in P^(i)
        Perform trust-region search with the start point x_k until its patience is reduced to zero, and denote the obtained optimum as x_k^*;
        Replace x_k with x_k^* in P^(i) in the spirit of Lamarckian learning;
    End for
End while
Return the best sample in the global database;
```

[1] All inputs and the corresponding responses obtained by solving the original expensive objective functions are archived into the global database.

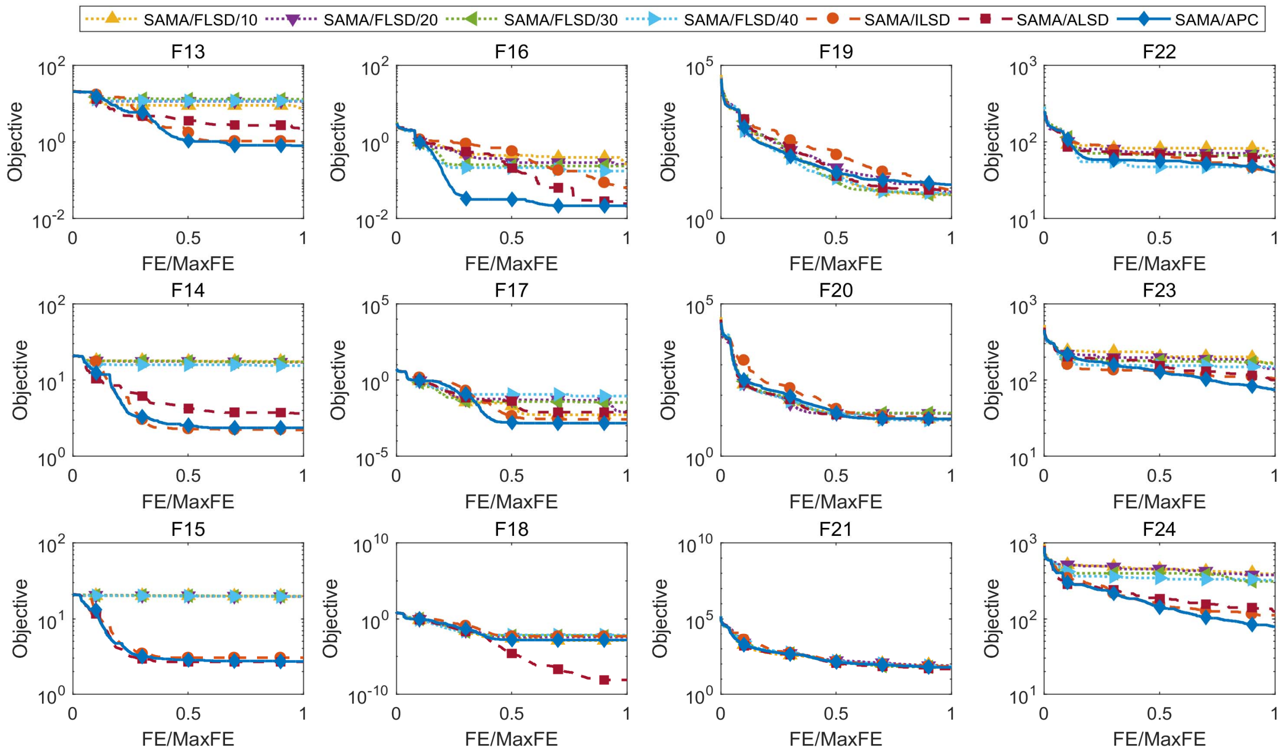

Fig. 3. Convergence curves of the mean objective values of the proposed SAMA/APC and the compared methods.

III. Experimental Studies

A. Test problems and settings

The CEC' 14 expensive optimization test suite [25] is employed to testify the performance of the proposed SAMA/APC. There are 24 test functions in the test suite, and only the 12 multimodal ones, i.e., F13 to F24, are used in the numerical experiments. The main features of the adopted 12 functions are summarized in Table II, and the detailed definition of those problems can be referred to the original paper [25].

The population size of the DE in the proposed SAMA/APC was set to 40. The mutation scaling factor F and the crossover control parameter CR were set to 0.8 and 0.4, respectively. The proposed method was implemented in MATLAB, and carried out on a personal computer with an Intel Core i5, 2.5GHz CPU. In order to reduce random variation in the numerical results and validate the robustness of the proposed method, 25 independent trials were performed for each test problem. The maximum number of function evaluations (MaxFE) for each test function was 50*D, where D is the dimension of the function.

The SAMAs with fixed, increasing, and adaptive LSD are employed as the compared methods. Four SAMAs with fixed LSD, denoted as SAMA/FLSD/10, SAMA/FLSD/20, SAMA/FLSD/30, SAMA/FLSD/40, respectively, are adopted, where the last number represents the maximum allowable depth of the local search in the SAMA. The SAMA with increasing LSD is denoted as SAMA/ILSD in the following discussions, in which the LSD starts from one and increases by one per generation. The SAMA with adaptive LSD is denoted as SAMA/ALSD, in which the LSD is dynamically adjusted according to the AFI method proposed in [14].

B. Result discussion

The convergence curves of the mean objective values over the 25 independent trials through the proposed SAMA/APC and the compared methods are plotted in Fig. 3. The mean and standard deviation of the optimal fitness values obtained by the SAMA/APC and the compared methods within the maximum number of FEs are listed in Table III and Table IV.

1) Comparison with fixed LSD methods

First, consider the comparison of the four SAMAs with fixed LSD. It can be observed from Fig. 3, Table III, and Table IV that:

- For problems F13, F14, F15, F19, F20, and F21, the SAMAs with different LSDs presents similar convergence trends;
- For problems F16, F22, F23, and F24, the convergence speed gradually improves as the adopted LSD increases;
- On the contrary, for problems F17 and F18, the increase of the LSD leads to the decrease of the convergence speed.

The above observations verify that the setting of the LSD has a significant impact on the optimization performance of SAMA. However, no consistent conclusion can be drawn from the above results of the four SAMAs with fixed LSD. Hence, in practice, it is difficult to provide an effective guideline on how to specify the fixed LSD to achieve better

TABLE III

EXPERIMENTAL RESULTS OF THE PROPOSED SAMA/APC AND THE COMPARED METHODS ON PROBLEMS F13 TO F18.

		F13	F14	F15	F16	F17	F18
SAMA/FLSD/10	Mean	7.330E+00	1.756E+01	1.976E+01	2.505E-01	5.333E-03	1.233E-03
	STD	4.743E+00	2.927E+00	5.761E-01	2.154E-01	8.298E-03	3.019E-03
SAMA/FLSD/20	Mean	1.155E+01	1.716E+01	1.964E+01	2.561E-01	7.597E-03	3.696E-03
	STD	7.696E-01	3.150E+00	1.894E-01	1.575E-01	7.652E-03	5.778E-03
SAMA/FLSD/30	Mean	1.308E+01	1.734E+01	1.981E+01	2.370E-01	3.459E-02	5.375E-03
	STD	3.930E+00	1.517E+00	1.444E-01	1.375E-01	5.064E-02	6.342E-03
SAMA/FLSD/40	Mean	1.177E+01	1.551E+01	1.958E+01	1.738E-01	8.765E-02	7.128E-03
	STD	3.406E+00	4.715E+00	2.126E-01	1.248E-01	1.602E-01	9.666E-03
SAMA/ILSD	Mean	1.040E+00	**2.220E+00**	3.033E+00	6.374E-02	2.734E-03	4.791E-03
	STD	9.272E-01	9.921E-01	7.369E-01	8.517E-02	7.676E-03	7.233E-03
SAMA/ALSD	Mean	2.199E+00	3.658E+00	**2.658E+00**	2.409E-02	7.802E-03	**7.753E-09**
	STD	5.300E-01	8.973E-01	6.087E-01	2.311E-02	1.015E-02	1.323E-09
SAMA/APC	Mean	**7.876E-01**	2.369E+00	2.715E+00	**2.154E-02**	**1.479E-03**	1.479E-03
	STD	9.775E-01	3.999E-01	1.523E+00	1.970E-02	3.308E-03	3.308E-03

TABLE IV

EXPERIMENTAL RESULTS OF THE PROPOSED SAMA/APC AND THE COMPARED METHODS ON PROBLEMS F19 TO F24.

		F19	F20	F21	F22	F23	F24
SAMA/FLSD/10	Mean	**5.822E+00**	1.675E+01	7.890E+01	6.487E+01	1.701E+02	3.780E+02
	STD	1.881E+00	1.605E+00	1.078E+01	2.343E+01	1.415E+01	3.507E+01
SAMA/FLSD/20	Mean	7.165E+00	2.512E+01	7.505E+01	6.463E+01	1.423E+02	3.791E+02
	STD	2.065E+00	2.295E+01	6.371E+01	1.438E+01	2.205E+01	4.774E+01
SAMA/FLSD/30	Mean	6.003E+00	2.602E+01	6.329E+01	6.609E+01	1.658E+02	3.101E+02
	STD	2.949E+00	2.062E+01	2.973E+01	2.941E+01	9.914E+00	3.191E+01
SAMA/FLSD/40	Mean	6.783E+00	**1.533E+01**	6.779E+01	4.731E+01	1.496E+02	3.225E+02
	STD	1.892E+00	5.012E+00	2.609E+01	1.949E+01	4.589E+01	7.633E+01
SAMA/ILSD	Mean	8.564E+00	1.635E+01	6.694E+01	4.085E+01	1.047E+02	1.114E+02
	STD	3.132E+00	3.946E+00	2.872E+01	1.588E+01	3.686E+01	4.332E+01
SAMA/ALSD	Mean	8.486E+00	1.671E+01	**4.655E+01**	4.790E+01	1.002E+02	1.249E+02
	STD	9.623E-01	1.304E+00	2.262E+01	2.066E+01	3.326E+01	5.352E+01
SAMA/APC	Mean	1.259E+01	1.698E+01	5.839E+01	**4.031E+01**	**7.450E+01**	**7.894E+01**
	STD	1.653E+01	2.983E+00	3.470E+01	2.213E+01	3.939E+01	2.379E+01

optimization performance, which demonstrates the disadvantage of the fixed LSD method.

Second, compare the results of the proposed SAMA/APC with those of the four SAMAs with fixed LSD. It can be seen from Fig. 3, Table III and Table IV that, for all test problems except F19, F20, and F21, SAMA/APC converges significantly faster than the four SAMA/FLSD methods. For F20 and F21, SAMA/APC presents similar convergence speed to the four SAMA/FLSD methods. Specially, for problem F19, SAMA/FLSD/10 and SAMA/FLSD/30 slightly outperform SAMA/APC.

The above results indicate that, in general, the proposed SAMA/APC significantly outperforms the SAMAs with fixed LSD. Besides, it is also possible that, if the LSD is carefully set to an appropriate value, the termination criterion based on fixed LSD can achieve better results than the proposed SAMA/APC. However, if no prior knowledge is available, it is computationally prohibitive or even impossible to run a trial-and-error procedure to select an appropriate LSD, especially for the computationally expensive optimization problems.

2) Comparison with variable LSD method

As shown in Fig. 3, Table III and Table IV, the following observations and conclusions can be made:

- The two SAMAs with variable LSD, SAMA/ILSD and SAMA/ALSD, are obviously superior to the four SAMA/FLSD methods in convergence speed for problems F13, F14, F15, F16, F23, and F24, which demonstrates the advantages of the variable LSD methods over the fixed LSD method.
- The SAMA/ILSD tends to converge slowly in the early stage of the optimization, and gradually accelerates the convergence later. This phenomenon can be found in the results of multiple test problems, including F13, F16, F19, and F22. This is because SAMA/ILSD starts with a small LSD and the local searches are prematurely terminated in a state of rapid decline, even without finding the local optima. Therefore, the individuals cannot be rapidly improved through local search. As the search proceeds, the increase of the LSD speeds up the convergence of SAMA/ILSD.

- For the extremely multimodal high-dimensional functions, F23 and F24, the proposed SAMA/APC, SAMA/ILSD, and SAMA/ALSD can continuously improve the current best optimum within the maximum number of FEs. This is because the three methods devote more computational efforts to global exploration at the beginning of the search process, and can effectively avoid the rapid decline in population diversity resulted from the Lamarckian learning.
- SAMA/APC can find much better solutions than SAMA/ILSD for F16, F23, and F24, but there is no test problem in the employed test suite, for which SAMA/ILSD can find significantly better solutions than SAMA/APC. Similarly, SAMA/APC outperforms SAMA/ALSD for F13, F14, F17, F23, and F24, but there is only one problem, i.e., F18, for which SAMA/ALSD find much better solutions than SAMA/APC. Therefore, these results demonstrate the superiority of the proposed SAMA/APC over the variable LSD methods.

IV. CONCLUSION

The commonly used termination criterion of the local search in SAMA is to set a fixed or variable LSD, which can lead to premature termination or excessive stagnation iterations of the local search. To address this problem, the SAMA/APC is proposed in this paper. In the framework of SAMA/APC, the DE is employed for global exploration; the proposed KPAS adaptively allocates a patience value to each individual of the current population; the TRS is performed on each individual for local exploitation, and terminates when the patience of the individual is reduced to zero.

The CEC' 14 expensive optimization test suite is employed to evaluate the performance of the proposed method, and the following conclusions can be drawn:

- When the computational budget is limited, the termination criterion of the local search has a significant impact on the optimization performance of the SAMA.
- The fixed LSD methods present the worst performance among the compared methods. Besides, in practice, it is difficult to pre-determine an appropriate setting for the fixed LSD, especially for computationally expensive optimization problems.
- The variable LSD method achieves much better results than the fixed LSD methods, and the proposed SAMA/APC slightly outperforms the variable LSD method, which verifies the efficiency of the proposed method.

In future studies, the proposed method will be further utilized for solving more industry-relevant problems, such as the aerodynamic optimization and structural optimization problems, to demonstrate its capability in real-world applications.

ACKNOWLEDGMENT

The present work was partially supported by the National Natural Science Foundation of China (grant no. 11502209, 51806175), and the National Defense Fundamental Research Funds of China (grant no. JCKY2016204B102, JCKY2016208C001).

REFERENCES

[1] V. G. Asouti and K. C. Giannakoglou, "Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes," *Engineering Optimization*, vol. 41, no. 3, pp. 241–257, Mar. 2009.

[2] Y. Lian, A. Oyama, and M.-S. Liou, "Progress in design optimization using evolutionary algorithms for aerodynamic problems," *Progress in Aerospace Sciences*, vol. 46, no. 5–6, pp. 199–223, Jul. 2010.

[3] L. Xia, L. Zhang, Q. Xia, and T. Shi, "Stress-based topology optimization using bi-directional evolutionary structural optimization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 333, pp. 356–370, May 2018.

[4] S. K. Azad, "Monitored convergence curve: a new framework for metaheuristic structural optimization algorithms," *Structural and Multidisciplinary Optimization*, Feb. 2019.

[5] W. E. Hart, N. Krasnogor, and J. E. Smith, "Memetic Evolutionary Algorithms," in *Recent Advances in Memetic Algorithms*, W. E. Hart, J. E. Smith, and N. Krasnogor, Eds. Berlin/Heidelberg: Springer-Verlag, 2005, pp. 3–27.

[6] P. T. H. Nguyen and D. Sudholt, "Memetic algorithms beat evolutionary algorithms on the class of hurdle problems," in *Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO '18*, 2018, pp. 1071–1078.

[7] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *AIAA Journal*, vol. 41, no. 4, pp. 687–696, Apr. 2003.

[8] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. 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.

[9] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-Assisted Cooperative Swarm Optimization of High-Dimensional Expensive Problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, Aug. 2017.

[10] W. E. Hart, "Adaptive global optimization with local search," university of California, 1994.

[11] D. Sudholt, "The impact of parametrization in memetic evolutionary algorithms," *Theoretical Computer Science*, vol. 410, no. 26, pp. 2511–2528, Jun. 2009.

[12] Y. Tenne and S. W. Armfield, "A framework for memetic optimization using variable global and local surrogate models," *Soft Computing*, vol. 13, no. 8–9, pp. 781–793, Jul. 2009.

[13] N. K. Bambha, S. S. Bhattacharyya, J. Teich, and E. Zitzler, "Systematic Integration of Parameterized Local Search Into Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 137–155, Apr. 2004.

[14] C. Liu and B. Li, "Memetic algorithm with adaptive local search depth for large scale global optimization," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 82–88.

[15] D. Molina, F. Herrera, and M. Lozano, "Adaptive Local Search Parameters for Real-Coded Memetic Algorithms," in *IEEE Congress on Evolutionary Computation*, 2005, vol. 1, pp. 888–895.

[16] J. Xu, C.-C. Wu, Y. Yin, and W.-C. Lin, "An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times," *Applied Soft Computing*, vol. 52, pp. 39–47, Mar. 2017.

[17] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[18] C. Jiang, D. Wang, H. Qiu, L. Gao, L. Chen, and Z. Yang, "An active failure-pursuing Kriging modeling method for time-dependent reliability analysis," *Mechanical Systems and Signal Processing*, vol. 129, pp. 112–129, Aug. 2019.

[19] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, "DACE-A MATLAB Kriging toolbox," 2002.

[20] Z.-H. Han and S. Görtz, "Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling," *AIAA Journal*, vol. 50, no. 9, pp. 1885–1896, Sep. 2012.

[21] D. R. Jones, "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, vol. 21, pp. 345–383, 2001.

[22] J. J. Moré, "Recent Developments in Algorithms and Software for Trust Region Methods," in *Mathematical Programming The State of the Art*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 258–287.

[23] Y. Yuan, "Recent advances in trust region algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 249–281, Jun. 2015.

[24] M. Bereta, "Baldwin effect and Lamarckian evolution in a memetic algorithm for Euclidean Steiner tree problem," *Memetic Computing*, vol. 11, no. 1, pp. 35–52, Mar. 2019.

[25] B. Liu, Q. Chen, Q. Zhang, J. J. Liang, P. N. Suganthan, and B. Y. Qu, "Problem Definitions and Evaluation Criteria for Computational Expensive Optimization," 2013.