A Surrogate-Assisted Clustering Particle Swarm Optimizer for Expensive Optimization Under Dynamic Environment

1st Yuanchao Liu
State Key Laboratory of Synthetical Automation for Process Industries Northeastern University
Shenyang, China
College of Information Science and Engineering Northeastern University
Shenyang, China
Yuanchaoliu@126.com

2nd Jianchang Liu*
State Key Laboratory of Synthetical Automation for Process Industries Northeastern University
Shenyang, China
College of Information Science and Engineering Northeastern University
Shenyang, China
liujianchang@ise.neu.edu.cn

3rd Tianzi Zheng
State Key Laboratory of Synthetical Automation for Process Industries Northeastern University
Shenyang, China
College of Information Science and Engineering Northeastern University
Shenyang, China
tianzizhengneu@163.com

4th Yongkuan Yang
State Key Laboratory of Synthetical Automation for Process Industries Northeastern University
Shenyang, China
College of Information Science and Engineering Northeastern University
Shenyang, China
yangyongkuanneu@sina.com

***Abstract*—In recent years, surrogate-assisted evolutionary algorithms have been developed for expensive optimization. However, a majority of applications are dynamic optimization problems in the real-world. In this paper, therefore, a surrogate-assisted clustering particle swarm optimizer is proposed for expensive dynamic optimization. In the proposed method, several clusters are first created by affinity propagation clustering, and then local radial basis function (RBF) surrogates are built based on the neighbor evaluated points for each cluster. Finally, in each cluster, the local RBF assists particle swarm optimizer to search the most promising point, which is evaluated by real objective function. To track dynamic environment, the points with best exact fitness in each cluster are added into new cradle swarm, if environmental change has occurred. A variety of experiments have been conducted on the moving peaks benchmark (MPB) with 500 change frequency in each environment. The experimental results have demonstrated that the proposed approach has a good performance.**

***Index Terms*—Expensive dynamic optimization, Surrogate-assisted, Particle swarm optimizer, Radial basis function, Affinity propagation clustering.**

I. Introduction

In most real-world optimization problems, the objective functions involve expensive experiments or simulations [1], such as several minutes or hours are required for the computational fluid dynamic (CFD) in a single evaluation [2], 36 to 160 hours are needed in a car crash simulation [3]. The above challenges pose a serious barrier for evolutionary algorithms (EAs) to tackle expensive optimization problems. Therefore, surrogate-assisted evolutionary algorithms (SAEAs), which can solve expensive optimization by employing budget surrogate models, have received a lot of attention [4]–[6]. In general, SAEAs can be roughly classified into two categories.

The first category mainly applies a single surrogate to assist EAs. For example, Jin *et al.* [7] developed a neural network-assisted EAs, and proposed an evolution control based on generation. In addition, the solutions with worse performance are filtered by radial-basis-function (RBF) networks, and the most promising solutions are exactly evaluated in [8]. In particle swarm optimizer (PSO), a local RBF model was built to predict the performance of each particle and a prescreening strategy was proposed [9]. Furthermore, Ong *et al.* [10] introduced a trust region method for the interleaved use of exact models for objective and constrained functions with a budget RBF in the local search. A Gaussian process (GP) model-assisted differential evolution (DE), which equipped with dimension reduction strategies for solving the "curse of dimensionality", was proposed for medium-scale computationally expensive optimization problems [11]. In recent years, a GP-assisted social learning PSO was proposed for high-dimensional expensive optimization, and an approximated fitness and uncertainty based multiobjective infill criterion was introduced [12]. Instead of applying the meta-model, a fitness estimation strategy (FES) was proposed to approximate fitness of point in PSO and competitive swarm optimizer [13], [14]. In addition, Pan *et al.* [15] applied an artificial neural network to predict the dominance relationship between candidate solutions and reference solutions for solving expensive many-objective optimization.

Multiple surrogates are employed for the second category. For example, Chugh *et al.* [16] proposed a surrogate-assisted reference vector guided evolutionary algorithm for expensive many-objective optimization, in which the local Kriging was adapted to approximate each objective function. Sun *et al.* [17] proposed a two-layer surrogate-assisted PSO, where the global model was built to smooth out the local optima and the

This work is supported by National Natural Science Foundation of China (61773106).

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

local model aims to provide an accurate search. A surrogate-assisted PSO based on ensemble model was developed [6], which adopted quadratic polynomial model, RBF network and Kriging. In addition, the FES-assisted PSO cooperates with RBF-assisted social learning-based PSO was proposed for high-dimensional expensive optimization [4]. Recently, Cai *et al.* [5], [18], [19] built a global surrogate model to prescreen the promising solution, while the local surrogate model in the neighbor region of each solution was introduced to guide mutation operator and update machine.

Despite the success of various literatures on SAEAs for expensive optimization, most of these methods do not consider dynamic environment. In fact, many real-world applications are dynamic optimization problems [20]. For example, dynamic shortest path routing problems [21], energy management system for hybrid electric vehicles [22] and dynamic scheduling problems [23]. Therefore, the aim of this work is to push the boundary of SAEAs by developing a surrogate-assisted clustering particle swarm optimizer (SA-CPSO) for expensive dynamic optimization problems. In the SA-CPSO, two strategies are introduced, namely, clustering technique and local RBF network-assisted PSO. The clustering technique aims to divide the initial cradle swarm into several small clusters, each of which searches the decision space simultaneously. It should be pointed out that affinity propagation clustering (APC) [24] is employed, which can avoid some extra parameters, e.g., the number of clusters or the cluster size. In addition, to solve the computationally expensive problems, the local RBF models are built for each cluster. Furthermore, in each cluster, the local RBF network assists PSO to search the most promising point, which is evaluated by real objective function. In order to adapt to a changing environment continuously, the points with best exactly fitness in each cluster are added to the new cradle swarm, if an environmental change has occurred.

The rest of this paper is organized as follows. Section II gives some related works. Section III provides the details of the proposed method. Extensive empirical studies are performed and discussed in Section IV. Finally, Section V concludes this paper.

II. Related Works

A. Particle Swarm Optimizer

The basic PSO, as a stochastic optimization method based on population, was first proposed by Kennedy and Eberhart [25], where the population is called a swarm and each point in swarm is called particle. In the search process, each particle is guided by the global best position found by the swarm and personal best position found by itself [26]. To be specific, a particle i has a velocity vector v_i and a position vector x_i, and the velocity vector v_i is updated as follows:

$$v_i' = \omega v_i + c_1 r_1 (x_{pbest_i} - x_i) + c_2 r_2 (x_{gbest} - x_i), \qquad (1)$$

where v_i' and v_i represent the after and before updated velocity of particle i, respectively, x_i is the current position of particle i, x_{pbest_i} and x_{gbest} are the best position found by particle i and global best position found by the swarm, respectively, ω is an inertia weight, c_1 and c_2 denote the acceleration constants, r_1 and r_2 are two different random numbers which generated from an uniform distribution in the range of 0 to 1. The position of particle i is updated by

$$x_i' = x_i + v_i', \qquad (2)$$

where x_i' represents the after updated position of particle i. Based on the update machine, the x_{pbest_i} will close to the x_{gbest} gradually. Thus, each particle can converge to the global best.

B. Radial Basis Function

RBF as the mate-model for SAEAs, it has robust and scalable characteristic, according to the comprehensive comparative studies conducted in [27]. Therefore, RBF is introduced to approximate real computationally expensive objective function in this paper.

The RBF networks was first proposed by Hardy [28], which is used to approximate the irregular surfaces. Given N different training data $X = \{x_1, x_2, \ldots, x_N\}$ and corresponding fitness values $F = \{f(x_1), f(x_2), \ldots, f(x_N)\}$, the interpolation form of the RBF is given as follows:

$$\hat{f}(x) = \sum_{i=1}^{N} \omega_i \varphi_i(\parallel x - x_i \parallel), \qquad (3)$$

where $\varphi_i(\bullet)$ denotes the i-th kernel function and $\parallel \bullet \parallel$ is the Euclidian norm. It should be pointed out that there are several types of basis kernel function, such as linear splines, thinplate splines, cubic splines, multiquadratics splines and Gaussian. In this paper, Gaussian function is selected, and $\varphi(x) = exp(-\frac{\|x-c\|}{\sigma^2})$. The parameters $\omega = \{\omega_1, \omega_2, \ldots, \omega_N\}$ represents the weight coefficients, which has the following form:

$$\omega = \Phi^{-1} F, \qquad (4)$$

where Φ denotes a $N \times N$ kernel matrix with $\Phi_{i,j} = \varphi(\parallel x_i - x_j \parallel)$.

III. Proposed Surrogate-Assisted Clustering Particle Swarm Optimizer

In this section, the main framework of the proposed method is first introduced, and then the details of SA-CPSO are given.

A. Main Framework of SA-CPSO

The general framework of SA-CPSO is summarized in Algorithm 1. The initial cradle swarm S with N points is first generated by Latin hypercube sampling (LHS) [29], and the fitness values F are evaluated using real expensive objective function. Then, S is automatically divided into n different clusters C by APC, which can avoid some extra parameters. Furthermore, in each cluster, PSO is performed on the local RBF for searching the most promising point. In addition, the overlapping check strategy is introduced to avoid multiple clusters searching an overlapping region. Finally, SA-CPSO will respond to change, if an environmental change has occurred. It should be pointed out that we assume that SA-CPSO will be informed when an environmental change happens, due to expensive computation for objective function. Additionally, the archive *DB* and *D* save the all evaluated points for each environment and all environments, respectively.

Algorithm 1 Main Framework of the SA-CPSO

```
Input: N (the initial population size), NL (the number of points for constructing local RBF);
Output: D (the exactly evaluated points for all environments);
Generate the initial cradle swarm S with N points by Latin Hypercube Sampling (LHS) and evaluate its real fitness F. Record the Pebst for each point and Gbest for each cluster;
DB ← [S, F]; //DB is an archive which saves all evaluated points //
(C, n) ← APC(S); //C concludes n clusters //
while stop criterion is not satisfied do
    for i = 1 to n do
        (C[i], DB, Pbest, Gbest[i]) ← local-RBF-assisted-PSO(NL, C[i], DB, Pbest, Gbest[i]);
    end for
    (C, DB, Gbest, Pbest) ← overlapping-check(C, n, DB, Gbest, Pbest);
    if The environment change has occurred then
        D ← [D,DB];
        SA-CPSO operates the change response strategy;
    end if
end while
```

B. Affinity Propagation Clustering

Affinity propagation clustering (APC) is applied for generating several clusters in the proposed method. In the APC, each particle can be viewed as a potential exemplar, and then subpopulations are created based on message-passing procedure [24].

To generate clusters, the correlations between each particle are used. In other words, the similarity $s(i,k)$, which indicates that whether particle k is an appropriate exemplar for particle i, should be determined at the initial stage. It should be noted that Euclidean distance not only can reflect the correlations between each particle, but also can be calculated easily in the search space. In this work, therefore, the negative Euclidean distance is introduced. As a result, the similarity $s(i,k)$ can be calculated as follows:

$$s(i,k) = - \parallel x_i - x_k \parallel^2, \tag{5}$$

where, x_i and x_k denote positions of particle i and particle j, respectively. It should be pointed out that the "preferences" $s(k,k)$ is set as the median of the similarities.

After calculating the similarity for each particle, APC will experience the massage-passing, which is a loop process. In the message-passing, the information between particles can be exchanged by two categories of messages, namely, "responsibility" and "availability". Based on the two messages, the message-passing can be performed by the following steps. To begin with, the initial value of availabilities $a(i,k)$ are set as zero. Next, the responsibilities can be calculated as follows:

$$r(i,k) = s(i,k) - max\{a(i,j) + s(i,j)\}, \tag{6}$$

where $j \neq k$. According to (6), the availabilities $a(i,k)$ are calculated by the following rule:

$$a(i,k) = min\left\{0, r(k,k) + \sum_{j \neq \{i,k\}} max\{0, r(j,k)\}\right\}. \tag{7}$$

However, the form of "self-availability" $a(i,i)$ is different from (7), which is given as follows:

$$a(i,i) = \sum_{j \neq \{i,k\}} max\{0, r(j,k)\}. \tag{8}$$

In addition, it is necessary for the "responsibility" and the "availability" to consider damping to avoid numerical oscillations, which may arise in some situations. Therefore, each message under the consideration of damping can be updated by the following:

$$r(i,k) = \lambda \times r(i,k)_{previous} + (1-\lambda) \times r(i,k), \tag{9}$$

$$a(i,k) = \lambda \times a(i,k)_{previous} + (1-\lambda) \times a(i,k), \tag{10}$$

where λ is the damping factor with range of 0 to 1, $r(i,k)_{previous}$ and $a(i,k)_{previous}$ denote the values from the previous iteration of $r(i,k)$ and $a(i,k)$, respectively.

The main steps of APC are listed below:

Step 1: Calculate the similarity $s(i,k)$ by (5), and initialize the availability $a(i,k) = 0$ and responsibility $r(i,k) = 0$;

Step 2: Calculate the values of responsibilities $r(i,k)$ by (6);

Step 3: If $i \neq k$, the values of availabilities $a(i,k)$ are given by (7); otherwise, the availabilities $a(i,k)$ are given by (8);

Step 4: Update the responsibilities $r(i,k)$ and availabilities $a(i,k)$ by (9) and (10), respectively;

Step 5: The point i can be viewed as the exemplar for point k, if $a(i,k) + r(i,k)$ achieves the maximum value for $i \in \{1,2,\ldots,N\}$.

Step 6: Repeat Steps 2 to 4 until the clustering termination criterion is satisfied.

Note that the termination criterion is that the APC experiences a fixed number of iterations fi or the estimated exemplars remain unchanged for some number of iterations Si. Therefore, the clusters can be created automatically based on the above process. In addition, the $\lambda = 0.5$, $fi = 200$ and $Si = 50$ as recommended by [24].

C. Local RBF-Assisted PSO

Since the objective function is computationally expensive problem, the local RBF models are built to approximate the fitness in each cluster. In order to build an accurate local RBF network, the training data should be selected appropriately. Therefore, $NL = 5 \times d$ (d denotes the variable dimensions) exactly evaluated points around the center of cluster are chosen as the training data. For example, given a cluster i, the center $center(i)$ of cluster i is given as follows:

$$center(i) = \frac{1}{|P|}\sum_{pi \in P} pi, \tag{11}$$

where P and $|P|$ represent the points in cluster i and the number of points in cluster i, respectively, pi denotes the position of each point in P. The Euclidean distance d between each point in the archive DB and $center(i)$ can be calculated, and the NL points with small d values are selected.

After constructing the local RBF models, the local RBF-assisted PSO is performed on each cluster to search the most promising point. In this search process, a generation-based model management strategy [30] is introduced. To be specific, in each cluster, the PSO performs G generations on the local RBF fitness landscape, and only the most promising point is exactly evaluated by the real objective function and added to the archive DB. It should be pointed out that, G is set as 100 in this work. Algorithm 2 gives the process of the local RBF-assisted PSO.

Algorithm 2 Local RBF-Assisted PSO

```
Input: $DB$ (the archive saves all evaluated points), $NL$ (the number of points for constructing local RBF), $C[i]$ (the $i$-th cluster), $Pbest$ (personal best for each point), $Gbest[i]$ (the global best of the $i$-th cluster);
Output: $DB$ (the archive saves all evaluated points), $C[i]$ (the $i$-th cluster), $Pbest$ (personal best for each point), $Gbest[i]$ (the global best of the $i$-th cluster);
1: Calculate the $center(i)$ of cluster $C[i]$ by (11);
2: Select the $NL$ training points according to the Euclidean distance between each point from $DB$ and $center(i)$ to construct the local $RBF$ network;
3: Perform PSO on the local RBF fitness landscape for $G$ generations, and the most promising point $p_m$ can be achieved;
4: $p_m$ is exactly evaluated by the real objective function;
5: Update the $C[i]$, $Pbest$ and $Gbest[i]$;
6: Add the point $p_m$ into the archive $DB$, $DB \leftarrow [DB,\ p_m]$;
```

D. Overlapping Check

In the SA-CPSO, multiple clusters may search an overlapping region in some situations. Therefore, the overlapping check strategy is introduced to avoid such phenomenon and effectively allocate computing resource for the clusters.

Generally speaking, the Euclidean distance between two different points can be easily calculated. Therefore, if the Euclidean distance between two global best points from different clusters is less than the threshold r_{excl}, it can be considered that the clusters are overlapping. Note that r_{excl} has the following form:

$$r_{excl} = 0.5\frac{X}{\sqrt[d]{n}}, \tag{12}$$

where X is the range of search space, d and n denote the dimension of variable and the number of clusters, respectively. In addition, the two clusters are merged into a cluster, while the other cluster will be re-initialized in the search space. The overlapping check is summarized in Algorithm 3.

Algorithm 3 Overlapping Check

```
Input: $C$ (the clusters), $n$ (the number of clusters), $DB$ (the archive saves all evaluated points), $Pbest$ (personal best for each point), $Gbest$ (the global best for each cluster);
Output: $C$ (the clusters), $DB$ (the archive saves all evaluated points), $Pbest$ (personal best for each point), $Gbest$ (the global best for each cluster);
Calculate the $r_{excl}$ by (12);
for $i = 1$ to $n$ do
    for $j = 1$ to $n$ do
        Calculate the Euclidean distance $d_{i,j}$ between $Gbest(i)$ and $Gbest(j)$ $(i \neq j)$;
        if $d_{i,j} < r_{excl}$ then
            if $f(Gbest(i))$ is better than $f(Gbest(j))$ then
                Merge $C[i]$ and $C[j]$ into $C[i]$, and remove the worst $\mid C[j] \mid$ particles from $C[i]$;
                Re-initialize the cluster $C[j]$;
                $DB \leftarrow [DB,\ C[j]]$;
            else
                Merge $C[i]$ and $C[j]$ into $C[j]$, and remove the worst $\mid C[i] \mid$ particles from $C[j]$;
                Re-initialize the cluster $C[i]$;
                $DB \leftarrow [DB,\ C[i]]$;
            end if
        end if
    end for
end for
Update the $Pbest$ and $Gbest$;
```

E. Change Response

Due to dynamic environment, it is necessary to apply the change response operation for SA-CPSO. If environmental change has occurred, the G_{best} in each cluster are preserved and re-evaluated by real objective function for the next environment. In order to maintain population diversity, a new cradle swarm with $N-n$ points are re-initialized. In addition, the re-evaluated G_{best} are added into the new cradle swarm, and APC is performed on the new cradle swarm to create new clusters. The archive DB saves the all evaluated points in the new environment, archive D saves the all evaluated points from each environment, and $Pbest$ and $Gbest$ are updated.

IV. Comparative Studies

A. Benchmark Problem

The moving peaks benchmark (MPB) is a classical dynamic test problem, which is first proposed by Branke [31]. In the MPB problem, the optima can be varied based on three features, e.g., the location, height, and width of the peaks. The

TABLE I
DEFAULT PARAMETERS FOR MPB

Parameter	Value	Parameter	Value
Peak shape	Cone	Number of dimensions, d	5
Number of peaks, p	10	Correlation coefficient, λ	0
Change frequency, U	500	Range of space for each dimension, S	[0, 100]
Environment, K	100	Range of height for each peak, H	[30.0, 70.0]
Shift length, s	1.0	Range of width for each peak, W	[1, 12]
Height severity, α	7.0	Initial height for each peak, I	50.0
Width severity, β	1.0		

MBP problem has the following form with a d-dimensional landscape:

$$f(x,t) = \max_{i=1,\dots,p} \frac{H_i(t)}{1+W_i(t)\sum_{j=1}^{d}(x_j(t)-X_{ij}(t))^2}, \tag{13}$$

where p is the number of peaks, $H_i(t)$ and $W_i(t)$ represent the height and width of peak i at time t, respectively, $X_{ij}(t)$ denotes the j-th dimension of peak i at time t. The position of each peak is shifted independently, and the move of a given peak i can be defined as follows:

$$\mathbf{v}_i(t) = \frac{s}{\mathbf{r}+\mathbf{v}_i(t-1)}((1-\lambda)\mathbf{r}+\lambda\mathbf{v}_i(t-1)), \tag{14}$$

where $\mathbf{v}_i(t)$ and $\mathbf{v}_i(t-1)$ denote the current and previous shift vector of peak i, respectively, s is shift length and determines the severity of the change, $\mathbf{r}$ is a random vector, λ is a correlation parameter in $[0,1]$. Afterthat, the position of peak i changes as follows:

$$X_i(t) = X_i(t-1)+\mathbf{v}_i(t). \tag{15}$$

In addition, the change of height and width of peak i can be described as follows:

$$H_i(t) = H_i(t-1)+\alpha*\sigma, \tag{16}$$

$$W_i(t) = W_i(t-1)+\beta*\sigma, \tag{17}$$

where α and β are height severity and width severity, respectively, and σ is a random number generated from a Gaussian distribution with mean 0 and variance 1.

Table I gives the default parameters for the MPB problem, where change frequency U and environment K indicate that environment changes every U evaluations and there are K environments for each run, respectively. It should be pointed out that the change frequency U is set as 500 in this work, because the expensive optimization problem is considered and the algorithms are required to converge to the optimum and adapt to the dynamic environment in the small number of evaluations.

B. Performance Indicator

The best-error-before-change E_{BBC} [32] is applied to validate the effectiveness of the proposed method, and the form of E_{BBC} is defined as follows:

$$E_{BBC} = \frac{1}{K}\sum_{k=1}^{K}(h_k-f_k), \tag{18}$$

where, K is the total environments, h_k is the best fitness for the k-th environment and f_k is the best fitness found by algorithm just before the k-th environment.

C. Comparison Methods and Parameter Settings

In this paper, the clustering particle swarm optimizer (CPSO) [33] and a variant KSA-CPSO are used for comparisons in empirical studies. CPSO is a classical dynamic optimization algorithm, and the best parameter settings are based on the corresponding paper. In KSA-CPSO, RBF network is replaced by Kriging surrogate and other components are same as SA-CPSO. The aim of designing this variant is to study the effect of different models, such as RBF and Kriging. In addition, the basic parameter settings of SA-CPSO and KSA-CPSO are given by Table II.

Each method is run for 25 times independently on the benchmark test problem, and the Wilcoxon rank sum test with a significance level of 0.05 is conducted to compare the significant difference between the compared methods. The best performance values are highlighted and shown in a gray background in the tables listing the comparison results.

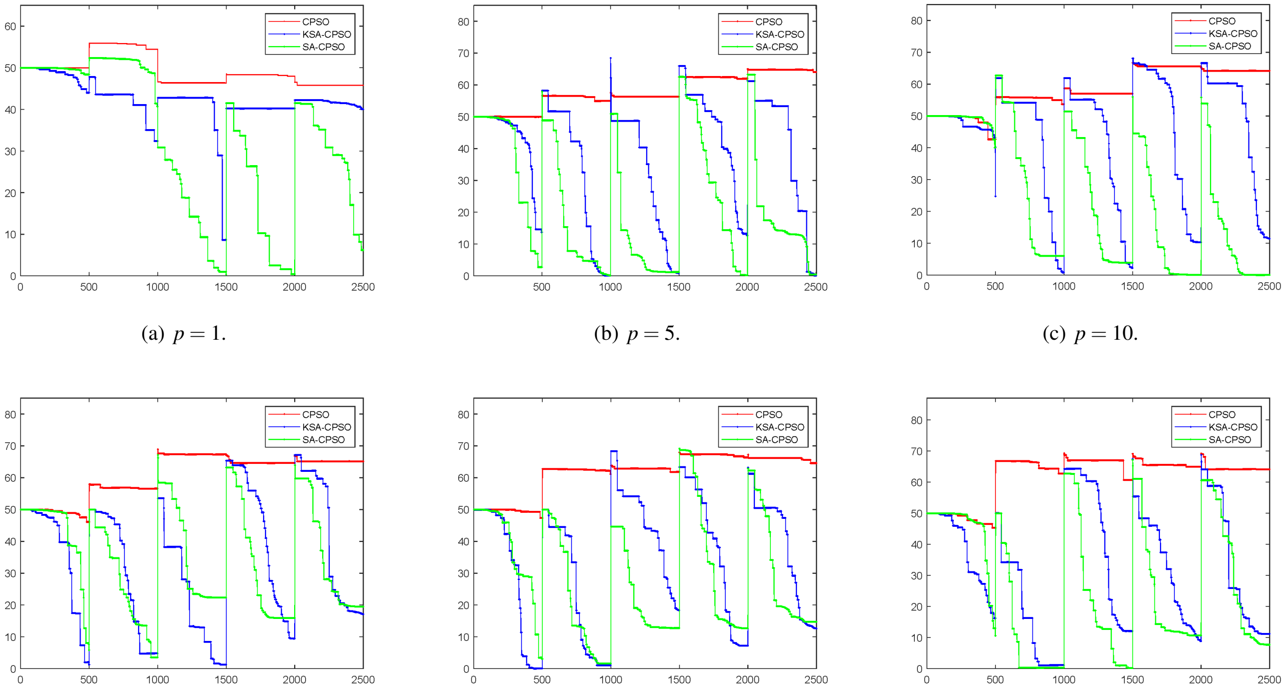
TABLE II
BASIC PARAMETERS

Parameter	Value	Parameter	Value
N	50	Inertia weight, ω	[0.3, 0.6]
fi	200	Acceleration constants, c_1, c_1	1.7
Si	50	Particle's velocity, v	[-5, 5]
λ	0.5		

D. Performance Comparison

Table III and Table IV give the best, worst, mean and the standard variance (Std.) values obtained by each algorithm on MPB with different number of peaks and different shift severities, respectively. The following observations can be made from the results presented in the tables.

First, the performance of CPSO is worse than that of other two surrogate-assisted methods on all dynamic environments. This is because that CPSO cannot converge to the optimum and fail to adapt to dynamic environment with 500 change frequency. In addition, this phenomenon indicates that the surrogate-assisted method can effectively solve expensive dynamic optimization problems.



(a) $p = 1$.

(b) $p = 5$.

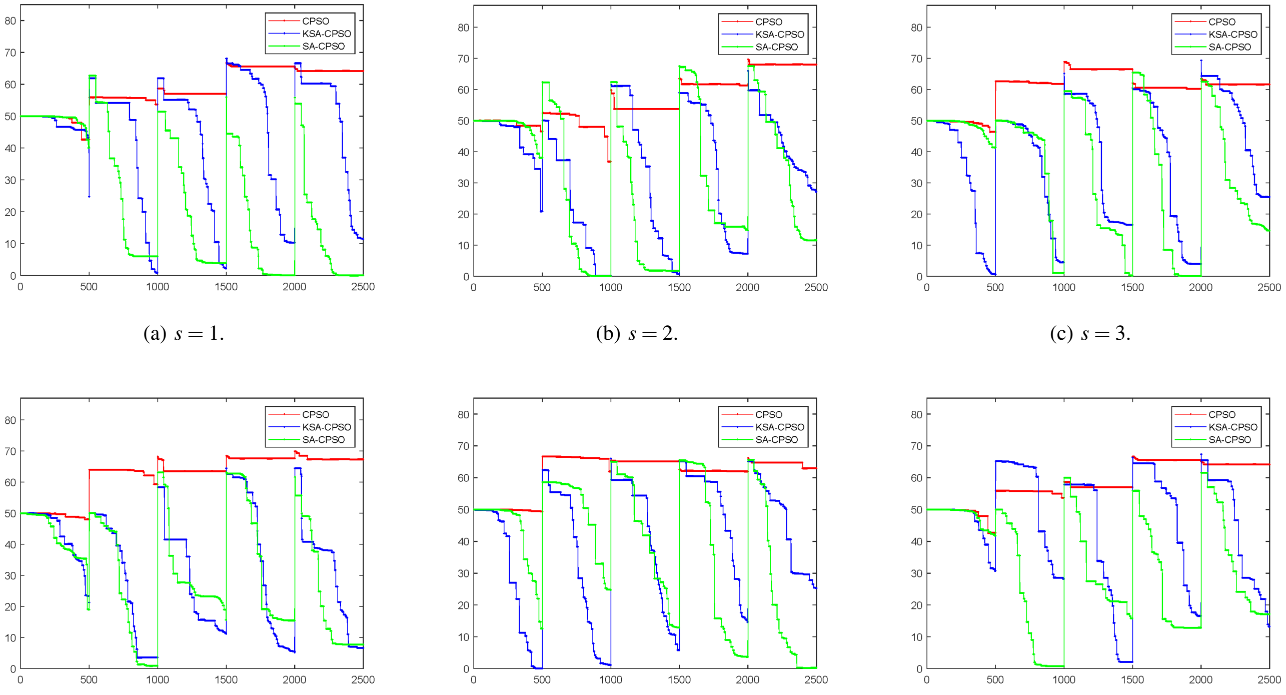
(c) $p = 10$.

(d) $p = 20$.

(e) $p = 30$.

(f) $p = 50$.

Fig. 1. Online errors of each method over fitness evaluations on MPB with different number of peaks.



(a) $s = 1$.

(b) $s = 2$.

(c) $s = 3$.

(d) $s = 4$.

(e) $s = 5$.

(f) $s = 6$.

Fig. 2. Online errors of each method over fitness evaluations on MPB with different different shift severities.

Second, the performance of SA-CPSO is better than that of KSA-CPSO on most situations. This can be attributed to the

TABLE III
THE STATISTICAL RESULTS OF THE E_{BBC} VALUES OBTAINED BY EACH METHOD ON THE MPB WITH DIFFERENT NUMBER OF PEAKS.

	Algorithm	Best	Worst	Mean	Std.	
$p = 1$	CPSO	40.3419	49.8987	45.4505	**2.6909**	+
	KSA-CPSO	8.7499	**17.8812**	13.8057	2.7842	+
	SA-CPSO	**6.4022**	18.5859	**12.0659**	3.2917	
$p = 5$	CPSO	46.8575	54.3706	50.0205	1.7134	+
	KSA-CPSO	5.8209	9.2179	6.9732	1.2805	+
	SA-CPSO	**3.2076**	**7.4191**	**4.4982**	**1.1703**	
$p = 10$	CPSO	46.1296	52.1557	48.7756	1.5515	+
	KSA-CPSO	5.3447	8.1992	6.6461	0.9062	+
	SA-CPSO	**3.2507**	**5.9493**	**4.5038**	**0.7909**	
$p = 20$	CPSO	42.8528	49.4592	45.7686	1.3885	+
	KSA-CPSO	5.8341	8.9711	7.3386	0.9152	+
	SA-CPSO	**4.2759**	**6.6804**	**5.2863**	**0.7075**	
$p = 30$	CPSO	41.2434	47.6429	43.7372	1.6481	+
	KSA-CPSO	6.1344	8.3319	7.1802	**0.6793**	+
	SA-CPSO	**4.3166**	**8.1555**	**6.3044**	1.0709	
$p = 50$	CPSO	39.0196	43.7392	41.7286	1.2574	+
	KSA-CPSO	6.8097	9.1187	8.0531	0.6992	+
	SA-CPSO	**4.8385**	**7.3224**	**6.0365**	**0.6542**	

TABLE IV
THE STATISTICAL RESULTS OF THE E_{BBC} VALUES OBTAINED BY EACH METHOD ON THE MPB WITH DIFFERENT SHIFT SEVERITIES.

	Algorithm	Best	Worst	Mean	Std.	
$s = 1$	CPSO	46.1296	52.1557	48.7756	1.5515	+
	KSA-CPSO	5.3447	8.1992	6.6461	0.9062	+
	SA-CPSO	**3.2507**	**5.9493**	**4.5038**	**0.7909**	
$s = 2$	CPSO	45.7789	51.5565	49.1845	1.4889	+
	KSA-CPSO	6.0558	8.8058	7.3733	**0.6936**	+
	SA-CPSO	**4.4214**	**7.6876**	**5.6743**	0.9672	
$s = 3$	CPSO	44.0964	52.4904	49.0721	2.0448	+
	KSA-CPSO	6.3923	10.5026	8.3238	1.0882	+
	SA-CPSO	**5.7035**	**8.8284**	**7.0603**	**1.0077**	
$s = 4$	CPSO	44.4633	52.2387	48.6397	1.6279	+
	KSA-CPSO	7.0866	10.5939	8.7991	1.1795	+
	SA-CPSO	**5.7741**	**9.1965**	**7.7994**	**1.1255**	
$s = 5$	CPSO	45.3139	51.7091	48.7479	1.6372	+
	KSA-CPSO	7.8381	**9.8743**	**8.2937**	**0.7383**	−
	SA-CPSO	**6.5286**	11.8116	8.7706	1.3522	
$s = 6$	CPSO	46.6743	52.4431	49.6792	1.4352	+
	KSA-CPSO	8.2853	**10.1555**	**8.7058**	**0.9928**	−
	SA-CPSO	**6.2752**	11.6571	9.3776	1.3801	

fact that the uncertainty of Kriging model are not considered by KSA-CPSO, and the Kriging fitness landscape may mislead the search process. In addition, the robustness of RBF is better than that of Kriging [27], when they are embedded in EAs.

Third, as the number of peaks p and shift severities s increase, the E_{BBC} values of SA-CPSO and KSA-CPSO arise. This is due to the fact that large p and s indicate that there are many local optimum and the location of each peak may move far away, respectively, which increase the difficulty of locating global optimum for algorithms. However, each method gets a large E_{BBC} value when $p = 1$, because multi-population methods are not suitable for single peak problem.

To further compare the convergence behaviors of different methods, the convergence profiles are plotted under different number of peaks and different shift severities in Fig. 1 and Fig. 2, respectively. From Fig. 1 and Fig. 2, it can be observed that the convergence of SA-CPSO is better than that of other methods and SA-CPSO can adapt to a changing environment continuously.

V. CONCLUSION

In this paper, a novel surrogate-assisted clustering particle swarm optimizer, termed as SA-CPSO, is proposed for expensive dynamic optimization. In SA-CPSO, an affinity propagation clustering is first introduced to create several clusters automatically. Then, in each cluster, local RBF assists PSO to search the most promising point, which is exactly evaluated by real objective function. Furthermore, overlapping check is adapted to prevent that multiple clusters search an overlapping region. To adapt to a changing environment continuously, the points with best fitness in each clusters are added to the new cradle swarm, if an environmental change has occurred. The comprehensive empirical experimental results have demonstrated that the surrogate-assisted method can effectively solve expensive dynamic optimization problems.

In the future, we will focus on employing other model management strategies for expensive dynamic optimization, such as probability of improvement (PoI) [34], expected improvement (ExI) [35] and lower confidence bound (LCB) [36]. In addition, we will investigate the effect of ensemble metamodels for expensive dynamic optimization algorithms.

REFERENCES

[1] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, June 2019.

[2] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, Aug 2009.

[3] T. Simpson, A. Booker, D. Ghosh, A. Giunta, P. Koch, and R.-J. Yang, "Approximation methods in multidisciplinary analysis and optimization: a panel discussion," *Structural and Multidisciplinary Optimization*, vol. 27, no. 5, pp. 302–313, Jul 2004.

[4] 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.

[5] X. Cai, L. Gao, and X. Li, "Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[6] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2664–2677, Sep. 2017.

[7] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, Oct 2002.

[8] M. K. Karakasis and K. C. Giannakoglou, "Metamodel-assisted multiobjective evolutionary optimization," *Eurogen*, 2005.

[9] C. Praveen and R. Duvigneau, "Low cost pso using metamodels and inexact pre-evaluation: Application to aerodynamic shape design," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 9, pp. 1087 – 1096, 2009.

[10] Y. Ong, P. Nair, and A. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[11] B. Liu, Q. Zhang, and G. G. E. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, April 2014.

[12] J. Tian, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Multiobjective infill criterion driven gaussian process-assisted particle swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 459–472, June 2019.

[13] C. Sun, J. Zeng, J. Pan, S. Xue, and Y. Jin, "A new fitness estimation strategy for particle swarm optimization," *Information Sciences*, vol. 221, pp. 355 – 370, 2013.

[14] C. Sun, J. Ding, J. Zeng, and Y. Jin, "A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 123–134, Jun 2018.

[15] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, Feb. 2019.

[16] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, Feb 2018.

[17] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Computing*, vol. 19, no. 6, pp. 1461–1475, Jun 2015.

[18] X. Cai, H. Qiu, L. Gao, C. Jiang, and X. Shao, "An efficient surrogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems," *Knowledge-Based Systems*, vol. 184, p. 104901, 2019.

[19] X. Cai, L. Gao, X. Li, and H. Qiu, "Surrogate-guided differential evolution algorithm for high dimensional expensive problems," *Swarm and Evolutionary Computation*, vol. 48, pp. 288 – 311, 2019.

[20] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "An affinity propagation clustering based particle swarm optimizer for dynamic optimization," *Knowledge-Based Systems*, vol. 195, p. 105711, 2020.

[21] S. Yang, H. Cheng, and F. Wang, "Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 1, pp. 52–63, 2010.

[22] J. T. B. A. Kessels, M. W. T. Koot, P. P. J. van den Bosch, and D. B. Kok, "Online energy management for hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3428–3440, Nov 2008.

[23] S. Yang and X. Yao, "Experimental study on population-based incremental learning algorithms for dynamic optimization problems," *Soft Computing*, vol. 9, no. 11, pp. 815–834, Nov 2005.

[24] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[25] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Oct 1995, pp. 39–43.

[26] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, May 1998, pp. 69–73.

[27] A. Díaz-Manríquez, G. Toscano, and C. A. Coello Coello, "Comparison of metamodeling techniques in evolutionary algorithms," *Soft Computing*, vol. 21, no. 19, pp. 5647–5663, Oct 2017.

[28] R. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of Geophysical Research*, vol. 76, pp. 1905–1915, 03 1971.

[29] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[30] H. Yu, Y. Tan, C. Sun, and J. Zeng, "A generation-based optimal restart strategy for surrogate-assisted social learning particle swarm optimization," *Knowledge-Based Systems*, vol. 163, pp. 14 – 25, 2019.

[31] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," vol. 3, 02 1999, p. 1882 Vol. 3.

[32] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1 – 24, 2012.

[33] S. Yang and C. Li, "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 959–974, Dec 2010.

[34] H. Ulmer, F. Streichert, and A. Zell, "Evolution strategies assisted by gaussian processes with improved preselection criterion," in *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, vol. 1, Dec 2003, pp. 692–699.

[35] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec 1998.

[36] M. T. M. Emmerich, K. C. 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, Aug 2006.