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

Similarity- and Reliability-Assisted Fitness Estimation for Particle Swarm Optimization of Expensive Problems

Tong Liu, Chaoli Sun, Jianchao Zeng, Songdong Xue and Yaochu Jin

Abstract—**As a population-based meta-heuristic technique for global search, particle swarm optimization (PSO) performs quite well on a variety of problems. However, the requirement on a large number of fitness evaluations poses an obstacle for the PSO algorithm to be applied to solve complex optimization problems with computationally expensive objective functions. This paper extends a fitness estimation strategy for PSO (FESPSO) based on its search dynamics to reduce fitness evaluations using the real fitness function. In order to further save the fitness evaluations and improve the estimation accuracy, a similarity measure and a reliability measure are introduced into the FESPSO. The similarity measure is used to judge whether the fitness of a particle will be estimated or evaluated using the real fitness function, and the reliability measure is adopted to determine whether the approximated value will be trusted. Experimental results on six commonly used benchmark problems show the effectiveness and competitiveness of our proposed algorithm. Preliminary empirical analysis of the search behavior is also performed to illustrate the benefit of the proposed estimation mechanism.**

I. Introduction

Particle swarm optimization (PSO), which is inspired by social behaviors such as bird flocking and fish schooling, was proposed by Kennedy and Eberhart in 1995 [1, 2]. PSO has received increasing attention for its easy implementation, fast convergence and strong global optimization capability. However, like other evolutionary algorithms such as Genetic Algorithms (GAs) and Differential Evolution (DE), PSO needs a lot of fitness evaluations to locate a global or near-global optimum, which prevents PSO from being applied to solve computationally expensive optimization problems.

The use of computationally efficient surrogate models to replace in part the expensive fitness evaluations is the most commonly adopted techniques to solve computationally expensive problems. Global or local surrogates using Gaussian Process (GP, also known as Kriging) [3], Support Vector Machine (SVM) [4] or Polynomial Regression (PR, also referred to response surface method) [5] have been proposed to substitute the computationally expensive real objective functions. Santana-Quintero et al. [6] presented a surrogate-assisted multi-objective PSO, where support vector machines (SVMs) are used as surrogates. To locate an optimum more quickly, Bird and Li [7] presented a technique through fitness approximation using regression. Gräning et al. [8] used an artificial neural network model trained online to approximate the objective function, which considerably reduces the computation time. Georgopoulou and Giannakoglou [9] and Gegis [10] introduced radial basis function networks (RBFNs) into PSO as a meta-model. In [9],the gradient value of the surrogate model was used to improve the convergence in the search of optimal solution. While in [10], the surrogate was used to identify the most promising trial position for each particle. In order to replace the exact computationally expensive objective functions during evolutionary search, Zhou et al. [11] proposed a novel surrogate-assisted evolutionary optimization framework that combines both global and local surrogate models. However, management of the surrogate in surrogate-assisted evolutionary algorithms is a challenging issue [12]. Based on the empirical convergence studies, Jin et al. [13] introduced individual- and generation-based evolution control to ensure the correct convergence of an evolutionary algorithm for optimization using an approximate fitness function, and proposed a framework for model management in generation-based evolution control. Alternatively, fitness inheritance techniques can be seen as a special local surrogate. In fitness inheritance, fitness evaluations can be reduced by estimating the fitness value of the offspring individuals from their parents [14]. Fonseca et al. [15] compared the impact to the evolutionary search introducing three inheritance techniques. A fitness estimation strategy was proposed by Sun et al. [16] for PSO, called FESPSO. Different from the standard fitness inheritance technique, the fitness of a particle in FESPSO is inherited not only from its parents, but also from its related individuals in previous generations and the current generation. In particular, in FESPSO, whether the fitness of a particle will be estimated is determined by its distance to other individuals in the current swarm whose fitness is known. The experimental results showed that the fitness estimation strategy in FESPSO has considerably enhanced the search performance of PSO. One weakness of the fitness estimation strategy in [1] is that the estimated fitness can be very inaccurate in some situations. This may deteriorate the effectiveness of the fitness estimation strategy, especially in the late stage of the search. In this paper, in order to enhance the accuracy in estimating the fitness, a similarity measure between different positions

T. Liu is with the Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China (e-mail: 1252160618@qq.com).
C. Sun is with the Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China (corresponding author, e-mail: clsun1225@163.com).
J. Zeng is with the Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China (e-mail: zengjianchao@263.net).
S. Xue is with the Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China (e-mail: xuesongdong@163.com).
Y. Jin is with the Department of Computing, University of Surrey, Guildford, GU2 7XH, United Kingdom (e-mail: yaochu.jin@surrey.ac.uk).

whose fitness will be used to estimate the fitness and a reliability measure on the estimated fitness are introduced to assess whether the fitness of a particle should be estimated or whether the estimated fitness is reliable.

The organization of the remainder of the paper is as follows. Section II gives a brief introduction to the PSO algorithm and the fitness estimation strategy in the FESPSO algorithm. Section III provides a detail description of the proposed similarity- and reliability-assisted fitness estimation strategy for PSO (SRFEPSO). Comparative results and empirical analysis of the search behavior of the SRFEPSO are presented in Section IV. Finally, Section V concludes the paper with a summary and ideas for future work.

II. PSO and the Fitness Estimation strategy in FESPSO

A. Particle swarm optimization

In PSO, a population (swarm) consists of particles with no volume and no weight, each of which represents a candidate solution to the optimization problem. Each particle in the swarm has its own velocity and position, and moves in a D-dimensional search space $S \in R^D$. The current velocity of each particle is determined by its previous velocity and the distance between the current position of particle and the position where this particle has achieved its best fitness so far (called the *personal best*), denoted by p_{best}, and to the position that has achieved the best fitness among all particles (called the *global best*), denoted by g_{best}. $\vec{v}_i(t) = (v_{i1}(t), v_{i2}(t), \ldots, v_{iD}(t))$ and $\vec{x}_i(t) = (x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t))$ are both D-dimensional vector representing the velocity and position of particle i at iteration t, respectively. Mathematically, the velocity and position of particle i are updated as follows [17]:

$$\begin{aligned}\vec{v}_{id}(t+1) =& \vec{v}_{id}(t) + c_1 r_1(p_{id}(t) - x_{id}(t)) + \\ & c_2 r_2(p_{gd}(t) - x_{id}(t))\end{aligned} \tag{1}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \tag{2}$$

where c_1 and c_2 are two positive constants referred to as cognitive and social parameters, respectively. r_1 and r_2 are two random numbers uniformly distributed in the range of [0,1]. $\vec{p}_i(t) = (p_{i1}(t), p_{i2}(t), \ldots, p_{iD}(t))$ is the personal best historical position found by particle i and $\vec{p}_g(t) = (p_{g1}(t), p_{g2}(t), \ldots, p_{gD}(t))$ is the global best historical position found by the whole swarm.

A number of variants of PSO have been proposed to improve the search performance of the algorithm. Two popular PSO variants are proposed by Shi (called the inertia weight model) [17] and Clerc (called the constriction factor model) [18]. In the inertia weight model, the velocity is updated as follows:

$$\begin{aligned}\vec{v}_{id}(t+1) =& \omega \vec{v}_{id}(t) + c_1 r_1(p_{id}(t) - x_{id}(t)) + \\ & c_2 r_2(p_{gd}(t) - x_{id}(t))\end{aligned}, \tag{3}$$

where ω is an inertia weight. In the constriction factor model of PSO, the velocity is updated as follows:

$$\begin{aligned}\vec{v}_{id}(t+1) =& \chi(\vec{v}_{id}(t) + \varphi_1(p_{id}(t) - x_{id}(t)) + \\ & \varphi_2(p_{gd}(t) - x_{id}(t)))\end{aligned} \tag{4}$$

where

$$\chi = \frac{2k}{2 - \phi - \sqrt{\phi^2 - 4\phi}}. \tag{5}$$

In (5), $\varphi_1 = c_1 r_1$, $\varphi_2 = c_2 r_2$, $\phi = c_1 + c_2$ and k is a real number in the range $(0, 1]$. Generally, $\phi > 4$, so c_1 and c_2 are usually set to 2.05.

The performance of PSO using inertia weight model and the constriction factor model have been compared by Eberhart and Shi [19], and their experimental results showed that PSO using the constriction factor and limiting the maximum velocity $v_{\max}$ to the maximum position $x_{\max}$ on each dimension performed the best. Therefore in this paper, PSO using the constriction factor is adopted to verify fitness estimation strategies.

B. The approximation strategy in FESPSO

FESPSO [16] is a PSO assisted by a fitness estimation strategy to reduce the evaluations using the computationally expensive real objective function. In FESPSO, the fitness of a particle is estimated using the following equation, which was deduced from the positional relationship between two arbitrary positions in the current swarm.

$$\begin{aligned}f(\vec{x}_j(t+1)) =& d_v^j(t+1)\Bigg\{\alpha[\frac{1}{d_v^i(t+1)} f(\vec{x}_i(t+1)) + \\ & \frac{1}{d_v^i(t_1)} f(\vec{x}_i(t-1)) \frac{1}{d_v^j(t)} f(\vec{x}_j(t)) + \\ & \frac{1}{d_v^{pj}} f(\vec{p}_j(t))] - (\frac{1}{d_v^j(t-1)} f(\vec{x}_j(t-1)) + \\ & \frac{1}{d_v^i(t)} f(\vec{x}_i(t)) + \frac{1}{d_v^{pi}(t)} f(\vec{p}_i(t)))\Bigg\}\end{aligned} \tag{6}$$

with

$$\alpha = \frac{\frac{1}{d_v^j(t+1)} + \frac{1}{d_v^j(t-1)} + \frac{1}{d_v^i(t)} + \frac{1}{d_v^{pi}(t)}}{\frac{1}{d_v^i(t+1)} + \frac{1}{d_v^i(t-1)} + \frac{1}{d_v^j(t)} + \frac{1}{d_v^{pj}(t)}}$$

where $d_v^j(t+1)$, $d_v^j(t-1)$, $d_v^j(t)$, $d_v^{pi}(t)$, $d_v^i(t+1)$, $d_v^i(t-1)$, $d_v^j(t)$ and $d_v^{pj}(t)$ represent the distance between $\vec{x}_v(t+1)$ and $\vec{x}_i(t+1)$, $\vec{x}_i(t-1)$, $\vec{x}_j(t)$, $\vec{p}_j(t)$, $\vec{x}_j(t+1)$, $\vec{x}_j(t-1)$, $\vec{x}_i(t)$ and $\vec{p}_j(t)$, respectively. All distances are calculated using the Euclidean distance. $\vec{x}_v(t+1)$ is a virtual position introduced for a better understanding of the fitness estimation strategy. Since the constriction factor model is adopted in this paper, the virtual position of is re-defined as follows according to the principle of the fitness estimation strategy proposed in [16]:

$$\begin{aligned}\vec{x}_v =& c_2\mathbf{r_2}\vec{x}_j(t+1) + c_2\mathbf{r_2'}(1 + \chi c_1\mathbf{r_1} - \chi c_2\mathbf{r_2})\vec{x}_i(t) \\ & + \chi c_2\mathbf{r_2}\vec{x}_j(t-1) + \chi c_1\mathbf{r_1} c_2\mathbf{r_2'}\vec{p}_i(t) \\ =& c_2\mathbf{r_2'}\vec{x}_i(t+1) + c_2\mathbf{r_2}(1 + \chi - \chi c_1\mathbf{r_1'} - \chi c_2\mathbf{r_2'})\vec{x}_j(t) \\ & + \chi c_2\mathbf{r_2'}\vec{x}_i(t+1) + \chi c_1\mathbf{r_1} c_2\mathbf{r_2}\vec{p}_j(t)\end{aligned} \tag{7}$$

where $\mathbf{r_1}$, $\mathbf{r_1'}$, $\mathbf{r_2}$ and $\mathbf{r_2'}$ are diagonal matrix whose diagonal elements are random number uniformly distributed in the range of [0,1].

III. A SIMILARITY- AND RELIABILITY-ASSISTED FITNESS ESTIMATION FOR PSO (SRFEPSO)

In FESPSO, once the fitness of a particle is known, either evaluated using the original fitness function or estimated, the fitness of its closest neighbor only will be estimated using Eq. (6). This may limit the number of particles whose fitness can be estimated. In order to further reduce the fitness evaluations using the computationally expensive objective function, an extension of FESPSO was proposed in [20], where a similarity measure was introduced so that all particles whose similarity to the particle whose fitness is known will be estimated. This has been shown to have effectively reduced the number of fitness evaluations using the real fitness function. However, the similarity based measure may bring about a weakness that there are too many particles whose fitness is estimated, which may mislead the search process. To address this weakness, in this paper, a reliability measure has been introduced into the fitness estimation strategy in order to maximize the number of particles whose fitness can be estimated while ensuring that the estimated fitness is as reliable as possible. In [20], the similarity was evaluated only between the particle whose fitness is known and others in the current swarm. However, it is insufficient in practice because positions other than the one in the current swarm are all participated in the fitness estimation in Eq. (6). Therefore a new similarity measure has been adopted in this paper, which is defined as the similarity between all positions used in Eq. (6). For two arbitrary positions $\vec{x}_i$ and $\vec{x}_j$, the new similarity measure is defined as follows:

$$s_{ij}(t+1) = 1 - \frac{\sqrt{\sum_{d=1}^{D}\left(x_{id}(t+1) - x_{jd}(t+1)\right)^2}}{\max\_\mathrm{dist}}, \tag{8}$$

where

$$\max\_\mathrm{dist} = max\{dist_d(t+1), d = 1, 2, \ldots, D\} \tag{9}$$

with

$$\begin{aligned}\mathrm{dist}_d(t+1) = & max\{x_{kd}, k = 1, 2, \ldots, n\} \\ & - min\{x_{kd}, k = 1, 2, \ldots, n\}\end{aligned} \tag{10}$$

where, $\vec{\mathrm{dist}} = (\mathrm{dist}_1, \mathrm{dist}_2, \ldots, \mathrm{dist}_D)$ represents the range of the current swarm.

It has been found that the estimated fitness can be very inaccurate, which may mislead the search process. The inaccuracy in similarity based fitness estimation strategy can be attributed to the fact that the fitness estimation in Eq. (6) may also be based on an estimated fitness of other particles. As a result, the use of similarity measure only to determine whether the fitness of a particle should be estimated may propagate the estimation error. To alleviate this problem, we propose to introduce a reliability measure for assessing the uncertainty of all estimated fitness. The basic idea for assessing the reliability of a fitness estimated using Eq. (6) is that the closer the current particle to a particle whose fitness is evaluated using the real fitness function and used for estimation, the higher the reliability. However, it is often not true when the fitness of particle is estimated. For this reason, the reliability of an estimated fitness should be evaluated differently as follows:

$$\begin{aligned} r_j(t+1) = \frac{1}{7}\Bigg(& \frac{r_i(t+1)}{dd_j^i(t+1)} + \frac{r_i(t-1)}{dd_j^i(t-1)} + \\ & \frac{r_j(t)}{dd_j^j(t)} + \frac{r_{pj}(t)}{dd_j^{pj}(t)} + \frac{r_j(t-1)}{dd_j^j(t-1)} + \\ & \frac{r_i(t)}{dd_j^i(t)} + \frac{r_{pi}(t)}{dd_j^{pi}(t)} \Bigg) \end{aligned} \tag{11}$$

where $r_j(t+1)$, $r_i(t+1)$, $r_i(t-1)$, $r_j(t)$, $r_{pj}(t)$, $r_j(t-1)$, $r_i(t)$ and $r_{pi}(t)$ are the reliability of the fitness at position $\vec{x}_j(t+1)$, $\vec{x}_i(t+1)$, $\vec{x}_i(t-1)$, $\vec{x}_j(t)$, $\vec{p}_j(t)$, $\vec{x}_j(t-1)$, $\vec{x}_i(t)$ and $\vec{p}_j(t)$, respectively. $dd_j^{pj}(t)$, $dd_j^i(t-1)$, $dd_j^j(t)$, $dd_j^{pj}(t)$, $dd_j^j(t-1)$, $dd_j^i(t)$ and $dd_j^{pi}(t)$ are the normalized distance between the current particle and those particles involved in fitness estimation, i.e., $\vec{x}_j(t+1)$ and $\vec{x}_i(t+1)$, $\vec{x}_i(t-1)$, $\vec{x}_i(t)$, $\vec{p}_j(t)$, $\vec{x}_j(t-1)$, $\vec{x}_i(t)$ and $\vec{p}_j(t)$. The normalization is performed simply by dividing the calculated distance between two particles by $\max\_\mathrm{dist}$ defined in Eq. (9). The sum of all weighted reliability measures is again divided by 7 to ensure that an estimated reliability always falls in [0, 1]. It should be noticed that if the fitness of a particle is evaluated using the real objective function, its reliability is set to 1.

Algorithm 1 summarizes the main steps of the proposed SRFEPSO algorithm.

Algorithm 1 The SRFEPSO algorithm

```
Initialize a population using Latin hypercube sampling method;
Calculate the fitness of each particle using the real objective function and set the fitness reliability of each particle to 1;
Determine the personal best historical position p_i for each particle;
Determine the best historical position of the swarm p_g;
While the stopping criterion is not met do
    Update the velocity and position of each particle in the swarm using Eq. (4) and (2);
    If it is the first time to update velocity and position then
        Calculate the fitness using the real objective function;
        Set the fitness reliability of each particle in the current swarm to 1;
    Else
        For each particle i in the swarm
            If its fitness is not known then
                Calculate the fitness using the real objective function;
                Set its fitness reliability to 1;
            End
            For each particle j, j ∈ {1, 2, ..., i−1, i+1, ... n} in the current swarm
                Calculate the similarity between an x_j(t+1) and each position in Eq. (6) using Eq. (8);
                Find the minimum similarity in all similarities, denoted by s_min,j
                Calculate the fitness reliability on position x_j(t+1) using Eq. (11);
                If s_max,j > δ1 and r_ij(t+1) > δ2 then
                    Estimate the fitness of particle j using Eq. (6);
                End;
            End For
            Determine the personal best historical position p_i for each particle;
```

End For
Determine the best historical position of the swarm $\vec{p}_g$;
If the fitness of $\vec{p}_g$ is estimated **then**
Calculate the fitness using the real objective function;
Compare the fitness of current $\vec{p}_g$ with the original one evaluated using the real objective function;
End If
End While

IV. Experimental Results and discussions

In order to evaluate the proposed SRFEPSO algorithm, six widely used benchmark problems are adopted in this paper. These include three unimodal test functions, Schwefel 1.2 (F1), Sum of Different Power function (F2) and Sphere Moxel (F3), and three multimodal test functions, Rasenbrock (F4), Rastrigin (F5) and Griewank functions (F6). The dimension of all test problems used in this work is set to $D = 30$. In our experiments, the thresholds of the similarity degree and the reliability degree are empirically set to 0.75 and 0.1, respectively. The stopping criterion is the maximum number of real fitness evaluations, which is set to 5000. All other parameters are set the same as those set in [16].

Table I shows the comparative results obtained by the PSO with constriction coefficient (CPSO), the FESPSO using constriction factor model (FESPSO) [16], the similarity-based FESPSO using constriction factor model (S-FESPSO) [20] and the proposed similarity- and reliability-assisted fitness estimation PSO (SRFEPSO). All compared algorithms are initialized using the Latin hypercube sampling method. Here, "Best", "Mean" and "Worst" stand for the best, the mean and the worst values of optimal solutions achieved in 30 independent runs, respectively. "Std." represents the standard deviation of the obtained optimal solutions.

TABLE I
Experimental Results on six test functions

Prob	Method	Best	Mean	Worst	Std.
F1	CPSO	2.01e+004	5.55e+004	1.34e+005	2.62e+004
	FESPSO	7.59e+003	5.06e+004	1.52e+005	3.38e+004
	S-FESPSO	8.17e+003	6.30e+004	1.25e+005	3.05e+004
	SRFEPSO	1.45e+004	**3.77e+004**	6.75e+004	1.32e+004
F2	CPSO	7.07e-016	1.48e-012	1.34e-011	3.21e-012
	FESPSO	3.34e-018	5.83e-014	1.65e-012	2.97e-013
	S-FESPSO	6.97e-110	**3.99e-085**	8.34e-084	1.63e-084
	SRFEPSO	1.79e-023	1.31e-016	1.98e-015	3.90e-016
F3	CPSO	2.64e-001	1.17e+000	3.99e+000	8.95e-001
	FESPSO	4.86e-002	4.78e-001	1.00+000	1.12e+000
	S-FESPSO	2.22e-008	**2.08e-002**	3.88e-004	1.01e-001
	SRFEPSO	2.80e-003	1.46e-001	7.20e-001	2.36e-001
F4	CPSO	3.82e+001	1.18e+002	2.10+002	4.64e+001
	FESPSO	1.34e+001	7.88e+001	2.14e+002	5.10e+001
	S-FESPSO	1.23e+001	8.77e+001	2.61e+002	5.21e+001
	SRFEPSO	1.72e+000	**6.18e+001**	1.38e+002	4.01e+001
F5	CPSO	2.47e+000	2.66e+001	4.23e+001	1.08e+001
	FESPSO	1.10e+000	2.27e+001	4.47e+001	1.29e+001
	S-FESPSO	2.20e-003	2.79e+001	5.33e+001	1.34e+001
	SRFEPSO	3.17e+000	**1.83e+001**	3.59e+001	1.18e+001
F6	CPSO	3.98e-001	8.05e-001	1.03e+000	2.05e-001
	FESPSO	5.74e-002	4.17e-001	1.02e+000	2.27e-001
	S-FESPSO	1.97e-001	7.16e-001	1.03e+000	2.27e-001
	SRFEPSO	1.36e-002	**1.47e-001**	5.30e-001	1.22e-001

From Table I, we can see that the proposed algorithm outperforms the three compared algorithms on F1, F4, F5 and F6. We can also see that S-FESPSO performs the best on F2 and F3.

A t-test of the optimization results obtained by the four compared algorithms has been performed and the results of the test with a significance level of 5% are listed in Table II. According to Table I and Table II, we can see SRFEPSO has achieved significantly better results than CPSO on all six problems. Compared to FESPSO and S-FESPSO, the proposed SRFEPSO performed significantly better on all three multimodal test problems. On the unimodal test problems, SRFEPSO achieved significantly better results on F3 than FESPSO, and on F1 than S-FESPSO, respectively.

TABLE II
Result of T-test with 5% Significance level comparing optimization results from SRFEPSO with other algorithms ("1" and "0" stand for dissimilarity and similarity, respectively)

	CPSO	FESPSO	S-FESPSO
F1	1	0	1
F2	1	0	0
F3	1	1	0
F4	1	1	1
F5	1	1	1
F6	1	1	1

The evolutionary profiles of CPSO, FESPSO, S-FESPSO and SRFEPSO are plotted in Figs. 1-6, where the X-axis represents the number of fitness evaluations and the Y-axis represents the fitness values averaged over 30 independent runs.

It can be easily seen that the SRFEPSO algorithm converges slowly at the first several iterations, and then more quickly than other PSO algorithms in the late search stage, which agrees with our conjecture. The reason is that the particles are scattered over the search space in the early stage of the search and consequently not many particles can satisfy the condition to estimate their fitness. In the late search stage, more and more particles will satisfy the condition for fitness estimation because they will move closer to each other as the swarm converges to a small region where an optimum is located. Therefore, the proposed SRFEPSO can obtain better results than the compared algorithms given the same number of real fitness evaluations.

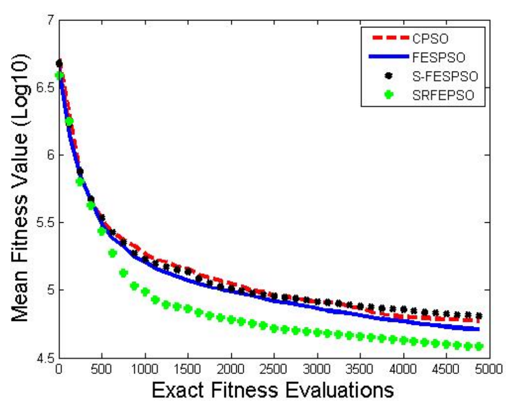

Fig. 1 The convergence profile on F1

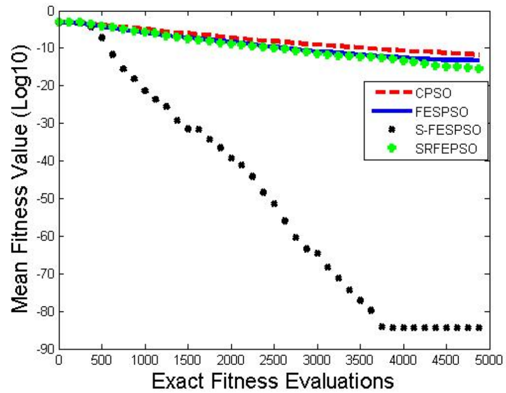

Fig. 2 The convergence profile on F2

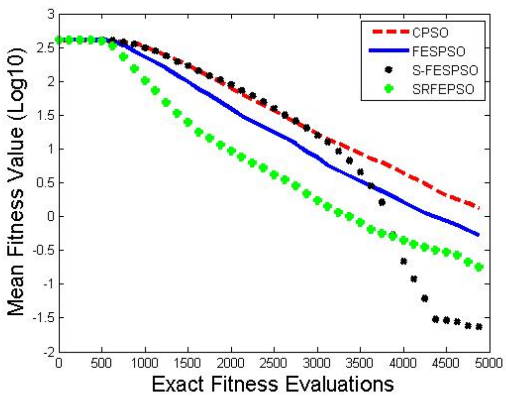

Fig. 3 The convergence profile on F3

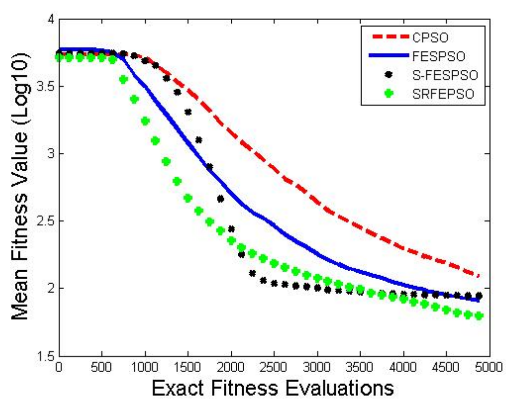

Fig. 4 The convergence profile on F4

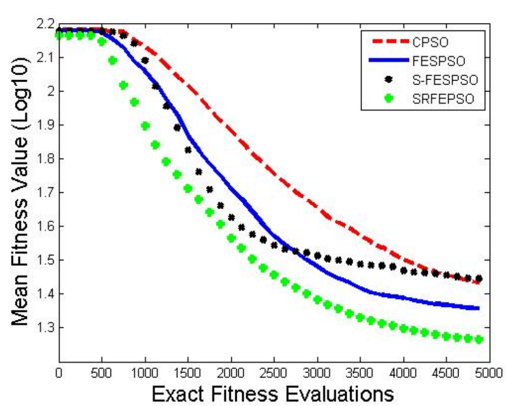

Fig. 5 The convergence profile on F5

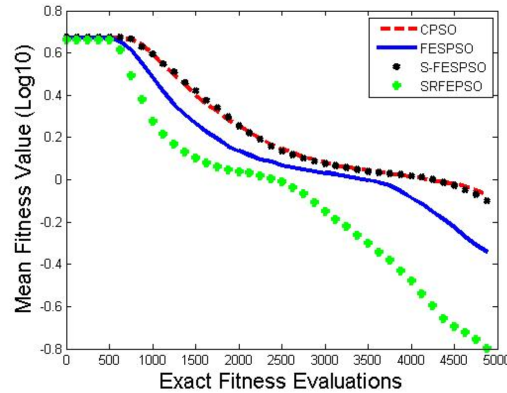

Fig. 6 The convergence profile on F6

In order to gain deeper insight into the proposed fitness estimation strategy to better understand the search behavior of the proposed SRFEPSO, we recorded the number of fitness estimations as well as the sum of estimation errors at each generation. The results are plotted in Fig. 7-8 and Fig. 9-10, respectively. We present the results from functions F2 and F4 in that the three compared PSO algorithms using fitness estimation have achieved comparable results on F2, while SRFEPSO has outperformed the other two PSO algorithms on F4.

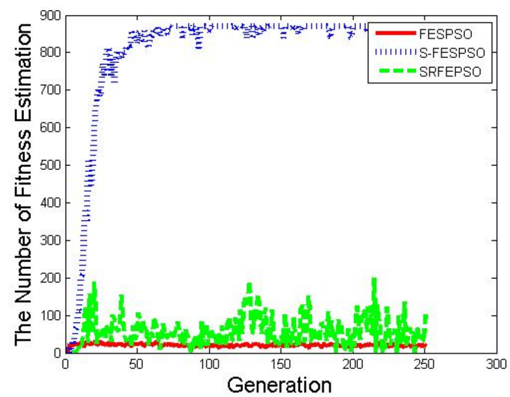

Fig. 7 The number of fitness estimations used in each generation for F2

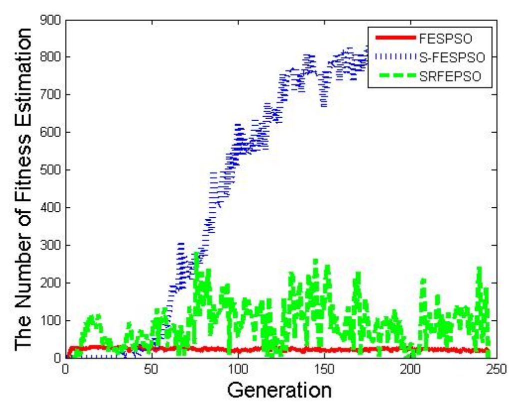

Fig. 8 The number of fitness estimations used in each generation for F4

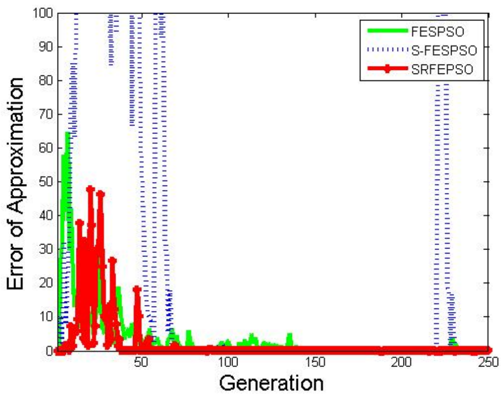

Fig. 9 Estimation errors of the three compared estimation strategies on F2

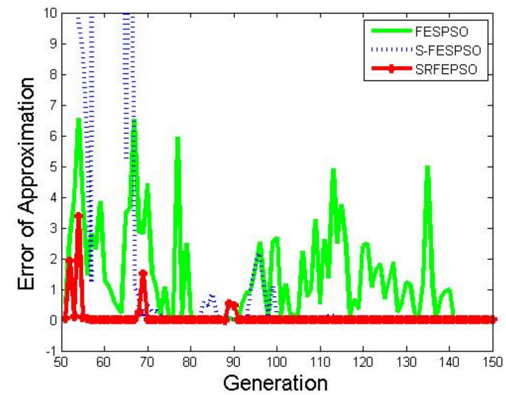

Fig. 11 The estimation errors profile on F2 at middle 100 generations

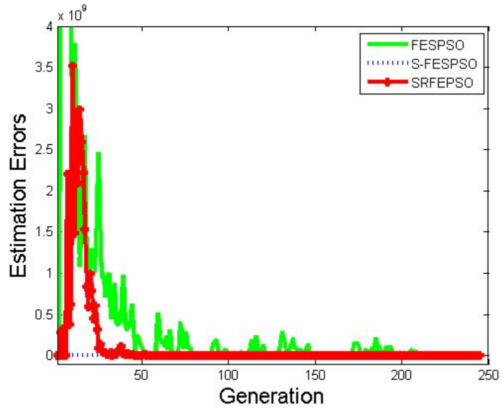

Fig. 10 Estimation errors of the three compared estimation strategies on F4

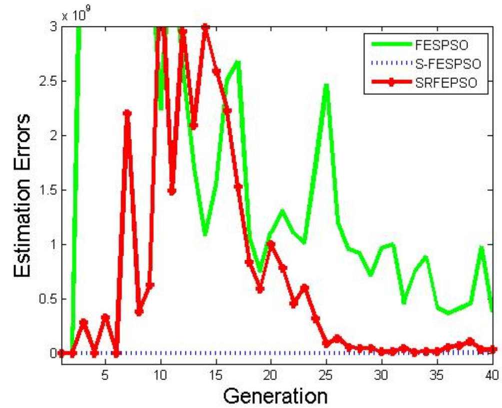

Fig. 12 The estimation errors profile on F4 at first 40 generations

From Figs. 7 and 8, we can find that S-FESPSO performed much more fitness estimations than FESPSO and SRFEPSO do for both functions, while SRFEPSO used much more fitness estimations than FESPSO. These results indicate that SRFEPSO can reduce the number of real fitness evaluations that FESPSO, but not as much as S-FESPSO. However, the optimization result of S-FESPSO on F2 is comparable to other algorithms and the result on F4 is poor than SRFEPSO. This means that more fitness estimations do not necessarily means better performance. Therefore, we further checked the accuracy of fitness estimations at each generation, which are shown in Figs. 9-10. From these results, we found that fitness estimations in SRFEPSO are more accurate than S-FESPSO in the early search stage. To get a closer look at the approximation errors in the late search stage, Fig. 11 provides a zoomed view of the estimation errors on F2 from generation 50 to generation 150, which clearly shows that the estimation accuracy of SRFEPSO is much higher than that of the other two estimation algorithms. Fig. 12 and Fig. 13 give a detail view on the estimation errors within the first 40 generations and those from generation 40 to generation 140, respectively. From these result, we can confirmed that the introduced reliability measure has effectively enhanced the fitness estimation accuracy. Note that in Fig. 12, the estimation errors of S-FESPSO are null as there is not fitness estimation, as shown in Fig.8. These preliminary empirical results suggest that accurate fitness estimations are important for enhance the search performance, particularly in the late search strategies.

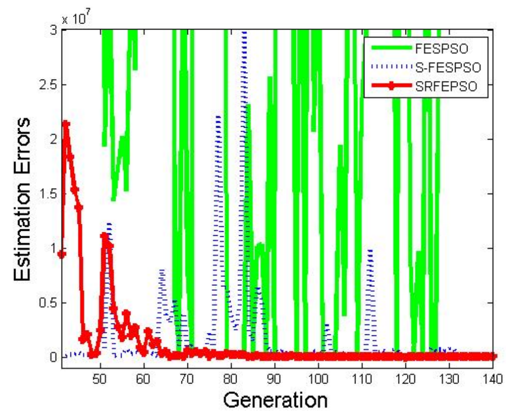

Fig. 13 The estimation errors profile on F4 at middle 100 generations

V. CONCLUSION

In this paper, in order to enhance the estimation accuracy while still reducing the real fitness evaluations as much as possible, a reliability measure has been introduced in combination with a similarity measure to determine whether the fitness of a particle should be estimated or calculated using the real fitness function. The experimental results demonstrate the effectiveness in that the proposed method, SRFEPSO, outperforms FESPSO and S-FESPSO on four out of six compared test functions. Empirical analysis of the number of fitness estimations used in the search process indicates that SRFEPSO used more fitness estimations than FESPSO but fewer than S-FESPSO. Meanwhile, the proposed algorithm achieved better estimation accuracy than the other compared fitness estimation strategies, which confirmed our assumption that the simultaneous use of similarity and reliability can help achieve a good balance between reducing the number of real fitness estimations and improving the estimation accuracy. This balance has led to an improved search performance on a majority of the studied test functions, in particular on multimodal test functions.

In the present work, the thresholds for reliability and similarity have been determined empirically. The influence of these thresholds on the search performance needs better understanding and further investigation. It might be interesting to introduce an adaptation mechanism for these thresholds to better control the exploration and exploitation during the search. Application of the proposed PSO algorithms using fitness estimation to real-world problems such as complex structure design will also be our future work.

ACKNOWLEDGMENT

This work was supported in part by Youth Foundation of Shanxi Province of China under Grant No. 2011021019-3, the Doctoral Foundation of Taiyuan University of Science and Technology under Grant No. 20122010, National Natural Science Foundation of China under Grant No. 6115016 and Shanxi Natural Science Foundation under Grant No. 2013011019-4.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," *in the IEEE International Conference on Neural Networks*, Washington DC, 1995, pp. 1942–1949.

[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

[3] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," *IEEE Transactions on Systems, Man, and Cybernetics*, *Part C: Applications and Reviews*, vol. 35, no.2, pp.183-194, 2005.

[4] L. Yang, D. Gong, X. Sun, J. Sun, "Surrogate models for user's evaluations based on weighted support vector machine in IGAs," *in 2012 10th World Congress on Intelligent Control and Automation (WCICA),* 2012, pp.144-149.

[5] B. Rutherford, "A response-modeling alternative to surrogate models for support in computational analyses," *Reliability Engineering & System Safety*, vol. 91, pp. 1322-1330, 2006.

[6] L.V. Santana-Quintero, et al, "Surrogate-based multi-objective particle swarm optimization," *Swarm Intelligence Symposium*, *SIS 2008. IEEE,* 2008, pp.1-8.

[7] S. Bird and X. Li, "Improving Local Convergence in Particle Swarms by Fitness Approximation Using Regression, "*in Computational Intelligence in Expensive Optimization Problems,* 2010, Springer, pp. 265-293.

[8] L. Graning, Y. Jin, and B. Sendhoff, "Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study," in *European Symposium on Artificial Neural Networks (ESANN' 2005)*, Bruges, 2005, pp.273-278.

[9] C. A. Georgopoulou and K. C. Giannakoglou, "A multi-objective metamodel-assisted memetic algorithm with strength-based local refinement," *Engineering Optimization*, vol. 41, no. 10, pp. 909-923, 2009.

[10] R.G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *Journal of Computational Science*, vol. 5, pp. 12-23, 2014.

[11] Zhou, et al., "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.

[12] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3-12, 2005.

[13] 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, 2002.

[14] R. E. Smith, B. A. Dike, and S. Stegmann, "Fitness inheritance in genetic algorithms," in *Proceedings of the 1995 ACM symposium on Applied computing*. 1995.

[15] L.G. Fonseca, A.C. Lemonge, and H.J. Barbosa. "A study on fitness inheritance for enhanced efficiency in real-coded genetic algorithms," In: *2012 IEEE Congress on Evolutionary Computation (CEC),* 2012, pp. 1-8.

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

[17] Y. Shi and R. Eberhart. "A modified particle swarm optimizer." in *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*, 1998, pp.69-73.

[18] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[19] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation,* 2000, vol. 1, pp.84-88

[20] C. Sun, J. Zeng, J. Pan and Y. Jin, "Similarity-based evolution control for fitness estimation in particle swarm optimization," in *2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE),* 2013, pp.1-8.