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

A Novel Improvement of Particle Swarm Optimization using Dual Factors Strategy

Lin Wang, Bo Yang*
Shandong Provincial Key Laboratory
of Network based Intelligent Computing,
University of Jinan, Jinan, 250022, China
*Corresponding author
Email: yangbo@ujn.edu.cn

Yi Li
School of Electrical Engineering,
University of Jinan,
Jinan, 250022, China

Na Zhang
Information Department,
China United Network
Communications Co. Ltd.
Shandong Branch,
Jinan, 250101, China

***Abstract*—The particle swarm optimization, inspired by nature, is widely used for optimizing complex problems and achieves many good stories in practical applications. However, the traditional PSO only focuses on the function value during evolutionary process. It ignores the information of distance between particles and potential regions. A Dual Factors Particle Swarm Optimization (DFPSO) incorporating both of distance and function information is proposed in this paper to help PSO in finding potential global optimal regions. The strategy of the DFPSO increases the diversity of population to yield improved results. The experimental results manifest that the performance, including accuracy and speed, are improved.**

I. Introduction

Traditional optimization algorithms cannot solve increasingly complex problems in reality because of their inflexible structure resulting from incomplete or noisy data and several multi-dimensional problems. A suitable solution to such problems is the nature-inspired optimization algorithm developed recently. One example of such algorithm is the particle swarm optimization (PSO) algorithm, which is based on the foraging of animals, such as birds. This algorithm has been widely utilized as an optimization tool in various applications ranging from communication, finance, and energy to medicine, materials science, and remote sensing [1]–[7].

The traditional PSO establishes two best solutions, which are viewed as temporary goals, and continually updates these solutions during the iteration process. The first one is the personal best solution, and the second one is the global best solution. All particles fly in the solution space following the two *bests* and gradually converges into the optimum area. However, traditional PSO focuses only on the function value during its evolution and not on the distance between the current positions and the potential global optimum area. The minimum value obtained so far does not guarantee that the area in which the current global best solution is located is the global optimum area. If the personal best and global best solutions are currently in the same local optimum area, traditional PSO may not be able to jump out of the local optimum area. Although the particles are located in the global optimum area, the aim is for the population to converge quickly and not oscillate between the global best and personal best solutions.

The above mentioned condition gives rise to this question: *can we consider both factors of distance and function value in searching multimodal regions thoroughly and avoiding falling into the local optimum area?* To address this question, a dual-factor PSO (DFPSO) that incorporates both distance and function information in helping PSO determine potential regions is proposed in this study. The strategy of DFPSO combines the information of distance and function value into a new *best* guiding particle to search the potential areas thoroughly. Thus, the population can jump out of the local optimum area via this bilateral behavior.

The rest of the paper is organized as follows. Section 2 provides a review of PSO. The details and analysis of DFPSO are shown in Section 3. Section 4 provides a discussion of the experimental results, and Section 5 presents the conclusion.

II. Particle Swarm Optimization

PSO, which was proposed by J. Kennedy et al. in 1995 [8], is a global numerical algorithm inspired by the foraging process of animals. From the perspective of the search space, the term "particle" represents a potential solution for a given problem. The population in PSO consists of a fixed number of particles. Each of the particles has two properties: velocity and position. Velocity decides the particle's flying direction and moving distance. The position decides where the particle is currently located in, that is, the solution to the given problem. In PSO, the particle moves in the search space and "chases" the best solution obtained so far. The position of the best particle will be decoded to the approximate optimum solution when the evolution ceases.

PSO emulates the forging behavior of a swarm of animals. Mathematically, the particle represents a point in the d-dimensional space. Two *bests* exist in PSO, namely, the historical best solution discovered by the particle itself (*pbest*) and the historical best solution identified by the population (*gbest*). These two solutions are tracked and updated by the particles accordingly. The PSO algorithm begins when a population of particles is initialized randomly and ceases when the termination criterion is satisfied. The entire searching process is preformed iteratively. The particle updates its new velocity as well as the position of the ith dimension as follows:

$$\begin{cases} v^i = v^i + \varphi_1 r(pbest^i - x^i) + \varphi_2 r(gbest^i - x^i) \\ x^i = x^i + v^i \end{cases} \quad (1)$$

where x represents a possible solution and v represents the velocity vector of the particle. r is a random positive number with a uniform distribution [0,1]. φ_1 and φ_2 are two user-defined acceleration constants. *pbest* is the best previous position that yields the best fitness value for the particle, and *gbest* is the best position discovered by the entire population. The complete traditional PSO algorithm is shown as Algorithm 1.

Algorithm 1: Algorithm of Particle Swarm Optimization

Input: Population size, acceleration constants φ_1, φ_2 and maximum velocity $VMAX$
Output: The best solution.

```
1  Initialize position x, velocity v for all of particles. The
   best position pbest and gbest are also initialized;
2  k=0;
3  while termination condition has not met do
4      Evaluate the fitness for each particle according to its
       position vector x;
5      Update the best position pbest for each particle;
6      Update the best position gbest for the whole
       population;
7      for id=1 to population size do
8          for i=1 to d do
9              v^i = v^i + φ1 r(pbest^i − x^i) + φ2 r(gbest^i − x^i);
10             v^i = min(VMAX^i, max(−VMAX^i, v^i));
11             x^i = x^i + v^i;
12         end
13     end
14 end
15 Return the best position.
```

A. Related Work

Many researchers have conducted numerous studies wherein velocity is restricted to help PSO search for the optimum solution carefully [9]–[18].

Shi and Eberhart [9]proposed the concept of inertia weight, which is widely utilized at present. The linearly decreased inertia weight, whose maximum and minimum are usually set to 0.9 and 0.4 [9], [10], respectively, decreases with the evolution of population. This method ensures the diversification of the population in the initial stage and fast convergence in the final stage. Researchers have also devoted much attention to the other components in velocity control, that is, the acceleration constants. Kennedy and Eberhart [8] suggested that velocity be set to a fixed value of 2.0. This setting is widely accepted and adopted in many applications. Ratnaweera et al. [17]. proposed a linearly time-varying acceleration constant-based PSO. A large φ_1 and a small φ_2 are set in the beginning and are gradually reversed during iteration to exchange the degrees of individual and social behaviors. Zhihui Zhan et al. [18] proposed an adaptive PSO by developing a systematic parameter adaptation scheme and an elitist learning strategy and by establishing an evolutionary state estimation technique. Several adaptive parameter control strategies are developed in their method.

Other researchers have attempted to use different methods that integrate the mechanisms of other optimization algorithms into PSO [19]–[25]. Lovbjerg et al. [19] integrated the mechanisms of evolutionary computation into PSO; subpopulations and breeding probability were introduced. Angeline [20] introduced the concept of selection derived from genetic algorithms to improve PSO. Their approach guarantees that the average quality of the particles is improved in every generation. Higashi and Iba [21] combined PSO with mutation, which is another concept in genetic algorithms. The updating rules in traditional PSO are improved through Gaussian mutation. Yang Shi et al. [22] proposed a cellular PSO in which a mechanism of cellular automata is integrated in the velocity update to avoid being trapped in the local optimum. El-Abd et al. [23] proposed a cooperative PSO algorithm based on migrating heterogeneous probabilistic models. Fan and Zahara [24] integrated the Nelder-Mead simplex method into PSO. Nakano et al. [25] improved the mechanism of PSO by tabu search.

The neighboring topology of a swarm, which is the flowing method of information, is also an issue for the improvement of PSO [26]–[33]. J. Kennedy et al. [26], [27] systematically investigated the effects of various topologies on PSO and tested four neighborhood topologies that guide the flight of particles. Frans van den Bergh et al. [31] proposed a cooperative particle swarm optimizer that exhibits cooperative behavior by using multiple swarms to optimize the different components of the solution vector. Mendes et al. [32] proposed a fully informed particle swarm (FIPS), which enables all neighbors to contribute to the update of velocity to make the individuals fully informed by the swarm. Their results show that FIPS is a promising method in searching the solution space. A dynamic multi-swarm PSO was proposed by J.J. Liang and P.N. Suganthan [33] to improve the topologies of the static neighborhood. This method divides the population into numerous, small, frequently regrouped subswarms among which information is exchanged.

J.J. Liang, A.K. Qin, and P.N. Suganthan et al. [34] explored a very different way and proposed a comprehensive learning particle swarm optimizer that optimizes every dimension of the solution vectors and enables the diversity of the swarm to be preserved to discourage premature convergence. A particle's velocity is updated by learning from all other particles' historical best information.

B. Challenge

Despite the success of traditional PSO in dealing with many problems, it only focuses on the *best*, which achieves the best approximate minimum/maximum and allows the particles to track the two *best*s. PSO ignores the distance between particles and the potential regions that include the real global optimum. Suppose that a particle is located in an optimum region and its current position is suboptimum among the population. If the current *gbest* and *pbest* (or *lbest* in local PSO) are far from the particle's position, the particle has to follow the instruction of these *best*s and is pulled to their regions. *gbest* and *pbest* easily cause the particle moving in its direction even if the particle is in a local optimum region far from the global optimum. Finally, the population converges into the local optimum.

DFPSO is proposed in this study to improve the performance of PSO by introducing a new *best*, which is called *ptbest*, and by combining the information of distance and

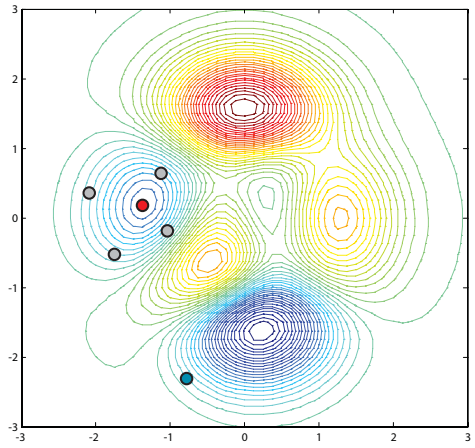

Fig. 1. The *bests* distributed in a two-dimensional searching space showed in contour plot. There are five particles in the swarm. The red point represents the position of current *gbest*. The others represent the *pbest* for each of particle. All of gray *pbest* and the red *gbest* are located in the local optimum area. The blue *pbest*, which is far away from the current *gbest*, is located in the global optimum area, but its function fitness is not the best one. According to the traditional particle swarm optimization, most of particles will be pulled toward the *gbest*, i.e. the local optimum area.

function value to obtain better results. Thus, DFPSO can jump out of the local optimum via a bilateral behavior. The following section provides details on DFPSO.

III. Dual Factors Particle Swarm Optimization

PSO is an optimization method that involves the discretization of the flight trajectory of particles. Related problems exist in the PSO strategy. First, discretization causes the particles to overlook the potential areas during the search process. Suppose that a particle is currently located in an area where the global optimum is also located and the particle is not merely located in the pinpoint of the global optimum. If the other particles detect a better point in the local optimum area, the particle in the global optimum area will be pulled toward the local optimum area. This condition results in premature convergence to the local optimum area. Second, if the entire population is located in the local optimum area far from the global optimum, traditional PSO may not be able to jump out of the local optimum area. Fig. 1 shows the disadvantages of traditional PSO. Therefore, the distance between *gbest* and the particle's personal memory should be considered in optimization to improve the search performance. If the *pbest* of a particle far from *gbest* ranks second only to *gbest* in fitness, this *pbest* has a high potential for helping the population search for the global optimum. The region search around *pbest* should be performed thoroughly.

DFPSO introduces a new *best* (*ptbest*) and adds it to PSO. *ptbest* is defined as the best position that has the highest *potential* in helping PSO identify the global optimum. *ptbest* combines factors from both the function value and distance. The critical measurement of *potential* has two targets: *pbest* that is located far from *gbest* and *pbest* that is better than most of the others. The different scales in the value of the two measures should be scaled to the same range first before being combined. Ranking-scaling method is adopted based on the diversification of optimized problems. *ptbest* is obtained

Algorithm 2: Algorithm of dual factors particle swarm optimization

Input: Population Size *PopSize*, and acceleration constants φ_0, φ_1, φ_2
Output: The best solution.
Initialization;
while *termination condition has not met* **do**
  Evaluate the fitness for each particle according to its position vector x;
  Update the best position *pbest* for each particle;
  Update the best position *gbest* for the whole population;
  for *j=1 to PopSize* **do**
    Calculate the euclidian distance between *pbest* of particle j and *gbest*, then record it into $Dis(j)$;
  end
  Sort *Dis* in the direction of ascend order, and record the ranking of each particle into *DisRanking*;
  Sort fitness of *pbest* for particles in the direction from worst to best, and record the ranking of each particle into *FitRanking*;
  for *j=1 to PopSize* **do**
    $Potential(j) = DisRanking(j) + FitRanking(j)$;
  end
  Update the best position *ptbest* for the whole population according to *Potential*;
  for *id=1 to PopSize* **do**
    for *i=1 to d* **do**
      $v^i = \omega v^i + \varphi_0 r(pbest^i - x^i) + \varphi_1 r(ptbest^i - x^i) + \varphi_2 r(gbest^i - x^i)$;
      $v^i = min(VMAX^i, max(-VMAX^i, v^i))$;
      $x^i = x^i + v^i$;
    end
  end
end
Return the best position found by all of particles.

by combining the rank of the distance between *pbest* and *gbest* and the rank of *pbest* for each particle. However, unlike *gbest* and *pbest*, *ptbest* only reflects the information of a potential region in the current generation. No updating of *ptbest* according to its historical value is implemented. The *potential* of the jth particle is calculated as follows:

$$Potential(j) = DisRanking(j) + FitRanking(j) \tag{2}$$

where $Potential(j)$ represents the *potential* of the jth particle, $DisRanking(j)$ represents the ranking of distance (from the nearest to the farthest) between *pbest* and *gbest* of the jth particle, and $FitRanking(j)$ represents the ranking of fitness (from worst to best) of the jth particle.

Accordingly, the velocity update of a particle is determined by the accumulated information from *pbest*, *gbest*, and *ptbest*. The velocity formula of the ith dimension of a particle is shown in the following equation.

$$\begin{cases} v^i = \omega v^i + \varphi_0 r(pbest^i - x^i) + \varphi_1 r(ptbest^i - x^i) + \varphi_2 r(gbest^i - x^i) \\ x^i = x^i + v^i \end{cases} \tag{3}$$

where *pbest* represents the best position determined by the particle itself, *ptbest* represents the best position with the highest *potential*, and *gbest* represents the best position identified by the entire population. A random inertia weight, $\omega = [0, \omega_{max}]$, is adopted in DFPSO to improve its search flexibility by restricting the change in velocity. Traditionally, setting ω_{max} to 1 is recommended for balance. φ_0, φ_1, and φ_2 represent the acceleration constant for each item. Algorithm 2 describes the details of DFPSO.

IV. Experiments and Results

To verify the performance of PSO in solving complex problems, DFPSO is tested based on eight benchmark functions

TABLE I. BENCHMARK FUNCTIONS

	Function	Search range	Best value	Fixed Accuracy Level
Unimodal Functions	$F_1 = \sum_{i=1}^{D} x_i{}^2$	$[-100,100]^D$	0	1.00E-06
	$F_2 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100,100]^D$	0	1.00E-06
	$F_3 = \sum_{i=1}^{D} i x_i{}^4 + random[0,1)$	$[-1.28,1.28]^D$	0	1.00E-06
	$F_4 = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100,100]^D$	0	1.00E-06
Multimodal Functions	$F_5 = \sum_{i=1}^{D-1} \left(100 \left(x_i{}^2 - x_{i+1} \right)^2 + \left(x_i - 1 \right)^2 \right)$	$[-2,2]^D$	0	1.00E-02
	$F_6 = -20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)} + 20 + e$	$[-32,32]^D$	0	1.00E-02
	$F_7 = \sum_{i=1}^{D} \left(y_i^2 - 10\cos(2\pi y_i) + 10 \right)$	$[-5,5]^D$	0	1.00E-02
	$F_8 = \sum_{i=1}^{D-1} \left(100 \left(z_i^2 - z_{i+1} \right)^2 + \left(z_i - 1 \right)^2 \right) + 390$ $z = x - o + 1$	$[-100,100]^D$	390	1.00E-02

TABLE IV. THE AVERAGE NUMBER OF FITNESS EVALUATIONS BEFORE REACHING THE FIXED ACCURACY LEVEL. THE BEST RESULTS AMONG THE SIX METHODS ARE HIGHLIGHTED IN BOLD AND DOT.

Dimension=10						
	GA	GPSO	LPSO	FIPS URing	FIPS Usquare	DFPSO
F1	1.26E+04	5.32E+04	6.28E+04	4.90E+04	1.94E+04	**•6.78E+03**
F2	3.77E+04	7.28E+04	1.00E+05	1.00E+05	4.38E+04	**•1.90E+04**
F3	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05
F4	**•3.31E+02**	3.69E+04	4.29E+04	1.99E+04	7.93E+03	2.43E+03
F5	1.00E+05	1.00E+05	1.00E+05	1.00E+05	1.00E+05	**•6.55E+04**
F6	9.00E+03	4.64E+04	5.49E+04	3.51E+04	1.25E+04	**•5.14E+03**
F7	1.00E+05	8.40E+04	1.00E+05	1.00E+05	1.00E+05	**•6.87E+04**
F8	1.00E+05	1.00E+05	9.70E+04	1.00E+05	1.00E+05	**•7.69E+04**
Dimension=20						
	GA	GPSO	LPSO	FIPS URing	FIPS Usquare	DFPSO
F1	4.09E+04	1.19E+05	1.44E+05	1.09E+05	2.85E+04	**•1.27E+04**
F2	2.23E+05	2.00E+05	2.00E+05	2.00E+05	1.49E+05	**•7.83E+04**
F3	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05
F4	7.27E+03	9.61E+04	1.13E+05	4.53E+04	1.20E+04	**•6.51E+03**
F5	1.97E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	**•1.45E+05**
F6	2.72E+04	1.07E+05	1.27E+05	7.69E+04	1.85E+04	**•8.94E+03**
F7	2.00E+05	**•1.97E+05**	2.00E+05	2.00E+05	2.00E+05	1.75E+05
F8	2.04E+05	2.00E+05	2.00E+05	2.00E+05	2.00E+05	**•1.57E+05**
Dimension=30						
	GA	GPSO	LPSO	FIPS URing	FIPS Usquare	DFPSO
F1	9.10E+04	1.92E+05	2.33E+05	1.83E+05	3.61E+04	**•1.93E+04**
F2	3.00E+05	3.00E+05	3.00E+05	3.00E+05	3.00E+05	**•2.03E+05**
F3	3.00E+05	3.00E+05	3.00E+05	3.00E+05	3.00E+05	3.00E+05
F4	1.21E+05	1.62E+05	1.98E+05	8.01E+04	**•1.52E+04**	2.05E+04
F5	2.89E+05	3.00E+05	3.00E+05	3.00E+05	3.00E+05	**•2.53E+05**
F6	6.02E+04	1.75E+05	2.09E+05	1.27E+05	2.34E+04	**•1.30E+04**
F7	3.00E+05	3.00E+05	3.00E+05	3.00E+05	3.00E+05	**•2.87E+05**
F8	3.00E+05	2.99E+05	3.00E+05	3.00E+05	3.00E+05	**•2.83E+05**

(four unimodal and four multimodal functions) [35], [36]. All the functions are tested on 10, 20, and 30 dimensions. Table I presents the properties and formulas of these functions.

Six algorithms, including the proposed DFPSO, four other PSO algorithms, and a genetic algorithm, are implemented. These algorithms are tested based on the eight benchmark functions with 10, 20, and 30 dimensions. The algorithms specifically include a genetic algorithm (GA), the global version of PSO with inertia weight (GPSO), the local version of PSO with ring neighborhood and inertia weight (LPSO), weighted fully informed particle swarm with U-ring topology (FIPS(URing)), weighted fully informed particle swarm with U-square topology (FIPS(USquare)), and the proposed DF-PSO. GA [37] is the most widely used among these methods. GPSO [8] and the LPSO with a ring topology [27] are traditional PSO algorithms. FIPS [32] is a fully informed PSO that employs all the neighbors to affect velocity. URing and USquare are employed with weighted FIPS based on goodness of fitness.

For a fair comparison, the maximum fitness evaluations are set to 100,000, 200,000, and 300,000 for 10-D, 20-D, and 30-D, respectively. The population size for all of the compared algorithms is set to 50. The parameters in the other five algorithms are set to their recommended best values. The value of acceleration constants φ_0, φ_1, and φ_2 in DFPSO is set to 1.2. To reduce statistical errors, each function is independently simulated 30 times; the best, worst, mean, and standard deviation are subsequently reported. The average number of fitness evaluations before reaching the fixed accuracy level is also reported for each problem. All experiments are performed in MATLAB 7.6 on the same machine equipped with AMD Phenom II 3.2GHz CPU, 3.1G memory, and Windows XP operating system.

Tables II and III provide the comparison of the accuracy of the proposed method and the other methods. The mean of DFPSO in most of the benchmark functions is better than those of the other methods. The smallest standard deviation is also generated by DFPSO. This result indicates that DFPSO is more stable than the other methods. DFPSO outperforms all other algorithms in functions F_1, F_2, F_3, F_5, and F_8 for the 10-D and 20-D problems and in functions F_1, F_2, F_3, F_5, F_7, and F_8 for the 30-D problem. DFPSO is tied with the other methods for the first rank in F_4 and F_6 for the 10-D and 20-D problems. Therefore, DFPSO significantly improves the optimization result in functions F_1, F_2, F_3, F_5, and F_8. Although the results achieved by DFPSO are not the best ones in 10-D and 20-D F_7, the best result is achieved by DFPSO in a higher dimension (30-D). DFPSO can thus be applied to solve complex problems. The dual-factor search strategy helps DFPSO improve its search performance. However, the results for F_6 exhibit a different trend. DFPSO is tied with FIPS

TABLE II. RESULTS ON 10-DIMENSIONAL AND 20-DIMENSIONAL PROBLEMS. THE BEST RESULTS AMONG THE SIX ALGORITHMS ARE HIGHLIGHTED IN BOLD AND DOT.

		F1	F2	F3	F4	F5	F6	F7	F8
Dimension=10									
GA	Best	4.42E-12	4.92E-10	4.91E-02	0.00E+00	2.49E-02	3.41E-06	2.00E+00	3.96E+02
	Worst	2.89E-10	4.84E-09	3.45E-01	0.00E+00	8.65E-01	2.94E-05	2.94E-05	1.78E+03
	Mean	1.10E-10	2.44E-09	1.25E-01	**•0.00E+00**	2.48E-01	1.11E-05	**•1.11E-05**	5.11E+02
	Std	6.88E-11	6.88E-11	6.24E-02	0.00E+00	1.89E-01	6.05E-06	6.05E-06	2.82E+02
GPSO	Best	3.88E-59	2.02E-20	3.00E-04	0.00E+00	1.90E+00	2.66E-15	0.00E+00	3.90E+02
	Worst	1.51E-53	5.25E-17	2.14E-03	0.00E+00	3.47E+00	2.66E-15	4.00E+00	4.84E+02
	Mean	1.02E-54	5.80E-18	9.42E-04	**•0.00E+00**	2.78E+00	2.66E-15	7.67E-01	4.07E+02
	Std	2.84E-54	1.28E-17	5.51E-04	0.00E+00	3.56E-01	0.00E+00	1.25E+00	2.53E+01
LPSO	Best	1.69E-28	9.51E-07	2.12E-04	0.00E+00	3.83E+00	6.22E-15	1.00E+00	3.90E+02
	Worst	1.29E-24	1.11E-04	4.18E-03	0.00E+00	4.96E+00	6.67E-13	5.00E+00	3.98E+02
	Mean	2.03E-25	1.96E-05	1.80E-03	**•0.00E+00**	4.47E+00	1.24E-13	3.42E+00	3.92E+02
	Std	3.17E-25	2.44E-05	8.51E-04	0.00E+00	3.53E-01	1.41E-13	1.30E+00	2.36E+00
FIPS_URing	Best	3.56E-18	1.81E-06	4.67E-04	0.00E+00	3.88E+00	5.43E-10	3.07E+00	3.92E+02
	Worst	4.35E-16	5.45E-05	2.43E-03	0.00E+00	4.54E+00	3.18E-07	6.92E+00	4.00E+02
	Mean	8.59E-17	1.24E-05	1.14E-03	**•0.00E+00**	4.18E+00	6.68E-08	5.33E+00	3.95E+02
	Std	8.44E-17	1.26E-05	5.10E-04	0.00E+00	1.72E-01	9.58E-08	8.48E-01	1.68E+00
FIPS_Usquare	Best	6.44E-49	2.24E-19	8.50E-05	0.00E+00	1.19E+00	0.00E+00	1.61E-02	3.91E+02
	Worst	9.48E-47	2.22E-16	1.36E-03	0.00E+00	2.05E+00	2.66E-15	6.14E+00	3.96E+02
	Mean	1.55E-47	3.15E-17	6.54E-04	**•0.00E+00**	1.63E+00	**•1.84E-15**	3.73E+00	3.92E+02
	Std	1.93E-47	5.31E-17	3.34E-04	0.00E+00	2.19E-01	1.53E-15	1.34E+00	1.24E+00
DFPSO	Best	1.93E-132	2.74E-48	7.94E-05	0.00E+00	3.51E-04	0.00E+00	0.00E+00	3.90E+02
	Worst	1.36E-124	4.38E-42	9.16E-04	0.00E+00	1.19E-03	2.66E-15	2.00E+00	4.06E+02
	Mean	**•5.33E-126**	**•5.42E-43**	**•3.17E-04**	**•0.00E+00**	**•5.86E-04**	**•1.84E-15**	4.00E-01	**•3.91E+02**
	Std	2.48E-125	1.06E-42	1.99E-04	0.00E+00	1.64E-04	1.53E-15	5.63E-01	3.09E+00
Dimension=20									
GA	Best	1.59E-10	3.17E-08	2.77E-01	0.00E+00	9.91E-06	1.64E-05	1.10E+01	3.90E+02
	Worst	2.20E-09	5.37E-06	1.18E+00	1.00E+00	1.10E+01	4.44E-05	2.10E+01	3.94E+02
	Mean	9.16E-10	7.13E-07	5.55E-01	3.33E-02	4.75E+00	2.56E-05	1.51E+01	3.91E+02
	Std	5.11E-10	1.36E-06	2.33E-01	1.83E-01	4.25E+00	7.07E-06	1.94E+00	1.83E+00
GPSO	Best	1.01E-52	4.31E-07	9.50E-04	0.00E+00	9.25E+00	2.66E-15	0.00E+00	3.90E+02
	Worst	1.40E-46	1.18E-04	5.53E-03	0.00E+00	1.63E+01	6.22E-15	9.00E+00	5.35E+02
	Mean	8.62E-48	3.47E-05	3.03E-03	**•0.00E+00**	1.22E+01	5.74E-15	**•2.63E+00**	4.17E+02
	Std	2.70E-47	3.09E-05	1.20E-03	0.00E+00	1.01E+00	1.23E-15	2.46E+00	3.55E+01
LPSO	Best	1.56E-21	5.45E-01	3.10E-03	0.00E+00	1.23E+01	1.45E-11	5.42E+00	3.90E+02
	Worst	2.82E-19	5.73E+00	1.02E-02	0.00E+00	1.47E+01	2.50E-10	1.90E+01	4.32E+02
	Mean	3.68E-20	2.42E+00	7.22E-03	**•0.00E+00**	1.42E+01	7.26E-11	1.33E+01	3.97E+02
	Std	6.72E-20	1.23E+00	2.02E-03	0.00E+00	5.08E-01	6.17E-11	3.54E+00	8.29E+00
FIPS(URing)	Best	6.85E-16	1.09E+00	1.10E-03	0.00E+00	1.33E+01	5.49E-09	1.71E+01	4.02E+02
	Worst	4.61E-15	7.11E+00	4.40E-03	0.00E+00	1.46E+01	3.25E-05	3.56E+01	5.15E+02
	Mean	2.40E-15	3.38E+00	2.83E-03	**•0.00E+00**	1.42E+01	2.12E-06	2.58E+01	4.11E+02
	Std	1.17E-15	1.66E+00	8.18E-04	0.00E+00	2.83E-01	7.24E-06	4.74E+00	2.11E+01
FIPS(USquare)	Best	1.85E-70	8.35E-11	7.18E-04	0.00E+00	9.57E+00	2.66E-15	9.98E+00	3.99E+02
	Worst	2.22E-68	8.03E-09	4.29E-03	0.00E+00	1.09E+01	2.66E-15	1.71E+01	4.06E+02
	Mean	3.43E-69	1.91E-09	1.97E-03	**•0.00E+00**	1.03E+01	**•2.66E-15**	1.37E+01	4.01E+02
	Std	5.43E-69	2.03E-09	8.95E-04	0.00E+00	2.99E-01	0.00E+00	1.62E+00	1.63E+00
DFPSO	Best	5.29E-147	1.67E-23	1.50E-04	0.00E+00	3.15E-05	2.66E-15	0.00E+00	3.90E+02
	Worst	9.84E-139	5.30E-19	3.66E-03	0.00E+00	6.10E+00	2.66E-15	1.30E+01	3.94E+02
	Mean	**•8.56E-140**	**•3.37E-20**	**•9.81E-04**	**•0.00E+00**	**•4.99E-01**	**•2.66E-15**	3.43E+00	**•3.91E+02**
	Std	1.99E-139	9.96E-20	7.27E-04	0.00E+00	1.46E+00	0.00E+00	3.47E+00	1.25E+00

(USquare) for the first rank in the 10-D and 20-D problems but ranks second in the 30-D problem. The result is slightly lower than that of FIPS (USquare).

The average number of fitness evaluations before reaching the fixed accuracy level in all the benchmark functions is counted and shown in Table. IV. Based on the results shown in the figure, DFPSO exhibits faster convergence speed compared with the other algorithms. The introduction of bilateral search behavior increases the possibility of determining a better position, that is, a better solution, with a small number of generations. Therefore, DFPSO improves the traditional PSO efficiently. Fig. 2 shows the comparison of evolutionary curve of errors on 30-dimensional problems. This plot further demonstrates that the advantages of DFPSO in accelerating evolution process as well as avoiding dropping into local optimum.

V. CONCLUSION

DFPSO was developed in this study to improve the performance of PSO. Both distance and function information were considered to help PSO identify the potential global optimum areas. The proposed strategy increases the possibility of jumping out of the local optimum area and yielding better results via the bilateral search behavior.

TABLE III. RESULTS ON 30-DIMENSIONAL PROBLEMS. THE BEST RESULTS AMONG THE SIX ALGORITHMS ARE HIGHLIGHTED IN BOLD AND DOT.

		Dimension=30							
		F1	F2	F3	F4	F5	F6	F7	F8
GA	Best	1.35E-09	1.80E-04	3.77E-01	0.00E+00	7.33E-04	2.73E-05	1.50E+01	4.78E+02
	Worst	9.70E-09	5.45E-03	3.18E+00	3.00E+00	6.78E+01	5.93E-05	3.30E+01	5.09E+02
	Mean	3.55E-09	1.27E-03	1.66E+00	5.00E-01	7.90E+00	4.02E-05	2.33E+01	4.94E+02
	Std	2.08E-09	1.18E-03	7.26E-01	7.31E-01	1.35E+01	7.39E-06	3.52E+00	6.96E+00
GPSO	Best	7.97E-45	4.20E-02	3.42E-03	0.00E+00	2.08E+01	6.22E-15	1.00E+00	3.90E+02
	Worst	3.45E-40	3.86E+00	1.11E-02	0.00E+00	2.77E+01	1.33E-14	2.80E+01	7.03E+02
	Mean	2.81E-41	5.56E-01	6.61E-03	**•0.00E+00**	2.25E+01	9.41E-15	1.20E+01	4.55E+02
	Std	7.02E-41	7.57E-01	2.28E-03	0.00E+00	1.71E+00	3.53E-15	6.94E+00	7.79E+01
LPSO	Best	1.35E-18	4.28E+01	9.14E-03	0.00E+00	2.21E+01	6.44E-10	1.30E+01	3.93E+02
	Worst	2.11E-15	1.75E+02	2.23E-02	0.00E+00	2.48E+01	1.64E-08	5.83E+01	5.71E+02
	Mean	1.71E-16	8.80E+01	1.56E-02	**•0.00E+00**	2.41E+01	3.45E-09	3.17E+01	4.39E+02
	Std	4.13E-16	3.78E+01	4.01E-03	0.00E+00	4.93E-01	3.24E-09	1.01E+01	4.15E+01
FIPS_URing	Best	6.23E-14	2.06E+02	2.72E-03	0.00E+00	2.38E+01	3.52E-08	5.26E+01	4.15E+02
	Worst	4.02E-13	1.14E+03	7.58E-03	0.00E+00	2.46E+01	6.07E-07	9.85E+01	7.71E+02
	Mean	1.70E-13	6.41E+02	4.96E-03	**•0.00E+00**	2.43E+01	1.06E-07	7.77E+01	4.46E+02
	Std	6.97E-14	2.18E+02	1.38E-03	0.00E+00	2.04E-01	1.12E-07	1.16E+01	7.11E+01
FIPS_Usquare	Best	4.44E-86	1.04E-06	1.36E-03	0.00E+00	1.90E+01	2.66E-15	2.07E+01	4.09E+02
	Worst	8.17E-84	3.86E-05	6.37E-03	0.00E+00	2.01E+01	6.22E-15	4.51E+01	5.65E+02
	Mean	1.93E-84	7.60E-06	3.74E-03	**•0.00E+00**	1.96E+01	**•3.85E-15**	2.78E+01	4.30E+02
	Std	2.45E-84	7.78E-06	1.22E-03	0.00E+00	3.08E-01	1.70E-15	4.43E+00	3.78E+01
DFPSO	Best	4.46E-145	4.41E-13	4.32E-04	0.00E+00	4.42E-05	2.66E-15	2.50E-12	3.90E+02
	Worst	7.35E-139	5.71E-10	1.05E-02	1.00E+00	7.66E+00	6.22E-15	2.40E+01	5.13E+02
	Mean	**•3.76E-140**	**•5.99E-11**	**•2.11E-03**	3.33E-02	**•8.27E-01**	4.09E-15	**•6.50E+00**	**•4.12E+02**
	Std	1.46E-139	1.16E-10	2.09E-03	1.83E-01	2.09E+00	1.77E-15	5.93E+00	4.09E+01

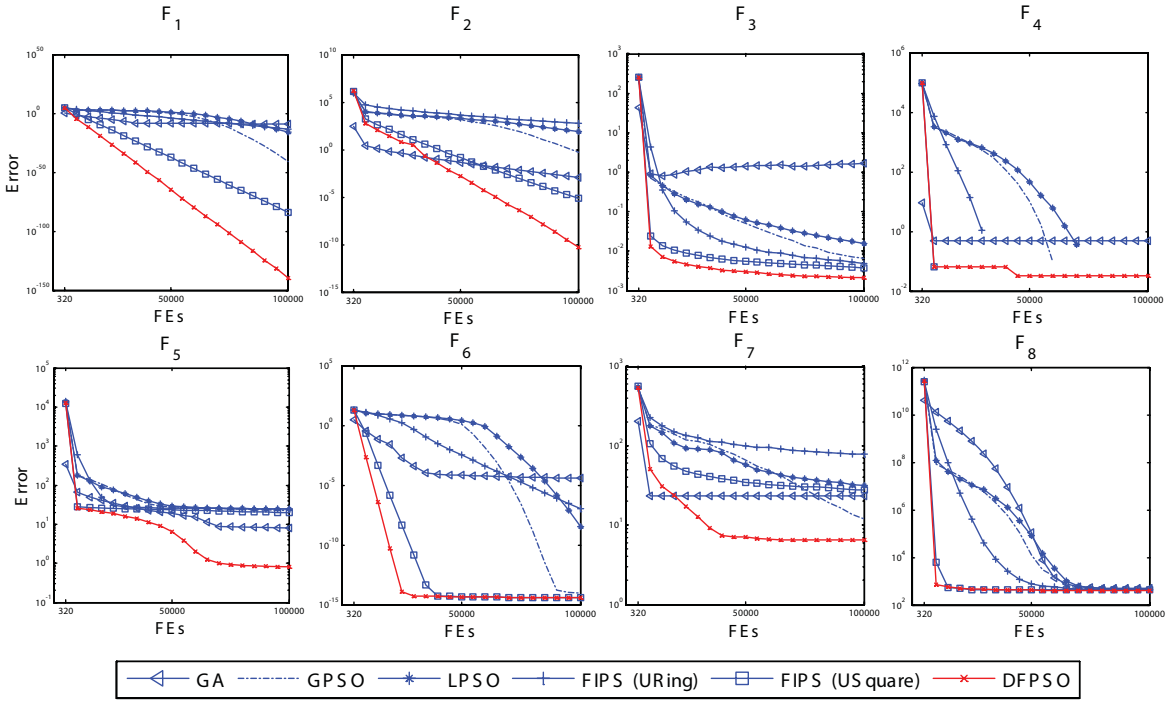

Fig. 2. The comparison of evolutionary curve of error on 30-dimensional problems.

Eight well-known benchmark functions were selected to compare DFPSO with five other algorithms, namely, four PSOs and a genetic algorithm. The experimental results show that the proposed approach improves performance, including accuracy and convergence speed.

ACKNOWLEDGMENT

This work was supported by National Key Technology Research and Development Program of the Ministry of Science and Technology under Grant 2012BAF12B07-3. National Natural Science Foundation of China under Grant No. 61173078, No. 61203105, No. 61373054, No. 61173079, No. 61070130, No. 81301298. Shandong Provincial Natural Science Foundation, China, under Grant No. ZR2010FM047, No. ZR2012FQ016, No. ZR2012FM010, No. ZR2011FZ001,No. ZR2010FQ028. Jinan Youth Science&Technology Star Project under Grant No. 2013012.

REFERENCES

[1] J. Ciurana , G. Arias , T Ozel , "Neural Network Modeling and Particle Swarm Optimization (PSO) of Process Parameters in Pulsed Laser Micromachining of Hardened AISI H13 Steel", *MATERIALS AND MANUFACTURING PROCESSES*, vol. 24, no. 3, pp. 358-368, 2009.

[2] Mahdiyeh Eslami , Hussain Shareef , Azah Mohamed , "Power system stabilizer design using hybrid multi-objective particle swarm optimization with chaos", *JOURNAL OF CENTRAL SOUTH UNIVERSITY OF TECHNOLOGY*, vol. 18, no. 5, pp. 1579-1588, 2011.

[3] Jui-Chung Hung, "Adaptive Fuzzy-GARCH model applied to forecasting the volatility of stock markets using particle swarm optimization", *INFORMATION SCIENCES*, vol. 181, no. 20, pp. 4673-4683, 2011.

[4] T.S. Lim , V.C. Koo , H.T. Ewe , et al. "A SAR autofocus algorithm based on particle swarm optimization", *Progress In Electromagnetics Research B*, vol. 1, pp. 159-176, 2008.

[5] H. A. Nguyen, H. Guo and K.-S. Low,"Real-Time Estimation of Sensor Node's Position Using Particle Swarm Optimization With Log-Barrier Constraint", *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3619-3628, 2011.

[6] Yamille del Valle , Ganesh Kumar Venayagamoorthy , Salman Mohagheghi , et al."Particle swarm optimization: Basic concepts, variants and applications in power systems", *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 12, no. 2, pp. 171-195, 2008.

[7] S.H. Zainud-Deen , W.M. Hassen , E.M. Ali et al, "Breast cancer detection using a hybrid finite difference frequency domain and particle swarm optimization techniques", *Progress In Electromagnetics Research B*, vol. 3, pp. 35-46, 2008.

[8] J. Kennedy and R. C. Eberhart, "A new optimizer using paritcle swarm theory", in *Proc. the Sixth Int. Symposium on Micromachine and Human Science*, pp.39-43, 1995.

[9] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer", *in Proc. IEEE World Congr. Comput. Intell.*, pp. 69-73, 1998.

[10] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization", *in Proc. IEEE Congr. Evol. Comput.*, pp. 1945-1950, 1999.

[11] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization", *in Proc. IEEE Congr. Evol. Comput.*, vol. 1, pp. 101-106, 2001.

[12] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms", *in Proc. IEEE Congr. Evol. Comput.*, pp. 94-97, 2001.

[13] M. Clerc, "The swarm and the queen: Toward a deterministic and adaptive particle swarm optimization", *in Proc. IEEE Congr. Evol. Comput.*, pp. 1951-1957, 1999.

[14] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space", *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58-73, 2002.

[15] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *in Proc. IEEE Congr. Evol. Comput.*, pp. 84-88, 2000.

[16] J. Kennedy, "The particle swarm social adaptation of knowledge", *in Proc. IEEE Int. Conf. Evol. Comput.*, pp. 303C308, 1997.

[17] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240-255, 2004.

[18] Zhi-Hui Zhan, Jun Zhang, Yun Li and Henry Shu-Hung Chung, "Adaptive Particle Swarm Optimization", *IEEE Transactions on Systems, Man, And Cyberneticslpart B: Cybernetics*, vol. 39, no. 6, pp. 1362-1381, 2009.

[19] M. Lovbjerg, T.K. Rasmussen, T. Krink, "Hybird particle swarm optimiser with breeding and subpopulations", *in: Proceedings of the GECCO*, pp. 469-476, 2001,

[20] P. Angeline, "Using selection to improve particle swarm optimization", *in: Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 84-89, 1998.

[21] N. Higashi, H. Iba, "Particle swarm optimization with Gaussian mutation", *in: Proceedings of IEEE Swarm Intelligence Symposium*, pp. 72-79, 2003.

[22] Shi Yang, Liu Hongcheng, Gao Liang, et al., "Cellular particle swarm optimization", *INFORMATION SCIENCES*, vol.181, no.20, pp.4460-4493, 2011.

[23] M. El-Abd, M.S. Kamel, "A cooperative particle swarm optimizer with migration of heterogeneous probabilistic models", *Swarm Intelligence*, vol. 4, no. 1, pp. 57-89, 2010.

[24] S.-K.S. Fan, E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization", *European Journal of Operational Research*, vol. 181, pp. 527-548, 2007.

[25] S. Nakano, A. Ishigame, K. Yasuda, "Particle swarm optimization based on the concept of Tabu search", *in: Proceedings of IEEE Congress on Evolutionary Computation*, pp. 3258-3263, 2007.

[26] J. Kenndy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance", *in: Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1931-1938, 1999.

[27] J. Kennedy, R. Mendes, "Population structure and particle swarm performance", *in: Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1671-1676, 2002.

[28] P.N. Suganthan, "Particle swarm optimiser with neighborhood operator", *in: Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1958-1961, 2002.

[29] K.E. Parsopoulos, M.N. Vrahatis, "UPSO-a unified particle swarm optimization scheme", *Lecture Series on Computational Sciences*, pp. 868-873, 2004.

[30] X.D. Li, "Niching without niching parameters: particle swarm optimization using a ring topology", *IEEE Transactions on Evolutionary Computation*, vol.14, no.1, pp.150-169, 2010.

[31] Frans van den Bergh, Andries P. Engelbrecht, "A Cooperative approach to particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, vol.8, no.3, pp.225-239, 2004.

[32] R. Mendes, J. Kennedy, J. Neves, "The fully informed particle swarm: simpler, maybe better", *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204-210, 2004.

[33] J. J. Liang and P. N. Suganthan,"Dynamic multi-swarm particle swarm optimizer", *in Proc. Swarm Intell. Symp.*, pp. 124-129, 2005.

[34] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", *IEEE TRANS. EVOL. COMPUT.*, vol.10, no.3, pp.281-295, 2006.

[35] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization". *Technical report, Nanyang Technological University, Singapore and KanGAL Report Number 2005005*, 2005.

[36] X.Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster", *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82-102, 1999.

[37] John H Holland, "Adaptation in Natural and Artificial Systems", *University of Michigan Press*, Ann Arbor, 1975.