Applied Soft Computing 26 (2015) 401–417

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems

A. Rezaee Jordehi *

Department of Electrical Engineering, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 5 December 2013
Received in revised form 19 July 2014
Accepted 17 October 2014
Available online 25 October 2014

Keywords:
Particle swarm optimisation
Global optimisation
Heuristics

ABSTRACT

Particle swarm optimisation (PSO) is a well-established optimisation algorithm inspired from flocking behaviour of birds. The big problem in PSO is that it suffers from premature convergence, that is, in complex optimisation problems, it may easily get trapped in local optima. In this paper, a new PSO variant, named as enhanced leader PSO (ELPSO), is proposed for mitigating premature convergence problem. ELPSO is mainly based on a five-staged successive mutation strategy which is applied to swarm leader at each iteration. The experimental results confirm that in all terms of accuracy, scalability and convergence rate, ELPSO performs well.

1. Introduction

A lot of real-world problems in different areas of sciences and technologies can be translated into optimisation problems. In literature, there exist various strategies for solving optimisation problems. Among them, particle swarm optimisation (PSO) is a heuristic optimisation algorithm that is being applied to different optimisation problems [1–3]. Some features of PSO can be listed as below.

- It has few parameters to be tuned by user [4].
- Its underlying concepts are quite simple and its coding is easier than other standard heuristic algorithms such as bacterial foraging optimisation algorithm (BFOA) and artificial bee colony (ABC).
- It does not require preconditions such as continuity or differentiability of objective functions.

As mentioned above, the main problem in PSO is that its exploration capability is weak. This problem makes its application in difficult optimisation problems, especially multimodal problems, problematic, so, devising new PSO variants with strong explorative capability is of high value and importance. Therefore, in this paper, the objective is to develop a new PSO variant with strong explorative and exploitative capabilities.

The rest of the paper is organised as follows; in Section 2, an overview of PSO is provided and some modified PSO variants are reviewed. In Section 3, the proposed PSO variant is introduced. Section 4 contains the experimental results and analysis. Finally, some conclusions are drawn in Section 5.

2. An overview of PSO and its modified variants

PSO is launched with initialisation of a swarm of particles in the n-dimensional search space (n is the dimension of problem in hand). In PSO, each particle keeps two values in its memory; its own best experience whose position and objective value are called P_i and P_{best} respectively and the best experience of the whole swarm, whose position and objective value are called P_g and g_{best} respectively. The position and velocity of particle i is represented by the following vectors:

$$X_i = (X_{i1}, X_{i2}, \ldots, X_{id}, \ldots, X_{in})$$

$$V_i = (V_{i1}, V_{i2}, \ldots, V_{id}, \ldots, V_{in})$$

The velocities and positions of particles are updated in each time step according to the following equations [5]:

$$V_{id}(t+1) = V_{id}(t) + C_1 r_{1d}(P_{id} - X_{id}) + C_2 r_{2d}(P_{gd} - X_{id}) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

* Tel.: +60 129102734.
E-mail address: ahmadrezaeejordehi@gmail.com

http://dx.doi.org/10.1016/j.asoc.2014.10.026
1568-4946/

Table 1
Some new PSO variants.

Ref.	Contribution	Validation
[12]	The global best solutions obtained by the PSO and ABC are used for recombination and the solution obtained from this recombination is given to the populations of the PSO and ABC as the global best and neighbour food source for onlooker bees, respectively.	Mathematical benchmarks and energy demand estimation problem
[3]	Newton's laws of motion are incorporated into PSO. It exposes desirable convergence speed, solution accuracy and global optimality.	Mathematical benchmarks
[13]	It employs the search logic of PSO but instead of having an actual swarm of solutions, makes use of a probabilistic representation of the population.	Mathematical benchmarks and power plant optimisation
[14]	It incorporates two strategies including a diversity enhancing mechanism and two neighbourhood search strategies.	Rotated multimodal and shifted mathematical benchmarks
[16]	Four updating strategies are combined in parallel to handle problems with different characteristics at different stages of the optimisation process.	Rotated, shifted and noisy mathematical functions and electric load dispatch problem
[17]	It uses multiple global best positions as elite examples to retain the diversity of the population.	Mathematical benchmarks
[15]	It hybridises cellular automata (CA) and PSO. A mechanism of CA is integrated in the velocity update to modify the trajectories of particles to avoid premature convergence.	Mathematical benchmarks
[18]	It hybridises PSO with gradient descent to utilise exploitation capability of gradient descent algorithm.	Mathematical benchmarks
[19]	It hybridises PSO with extremal optimisation (EO) to utilise EO's exploitative capability.	A few mathematical benchmarks
[20]	After updating particles by flight equations, half of the particles are affected by GA operators.	Optimal allocation of active power line controller in power systems.
[21]	Inertia weight of the most fitted particle is set to minimum and for the lowest fitted particle takes maximum value. So, the most fitted particle moves slowly in comparison to the lowest fitted particle. The most fitted particle takes rank first and its weight will be less while, the lowest fitted takes maximum rank their weights will be high.	Short term power generation scheduling
[22]	It applies a new flight equation wherein which phase angles are used instead of velocity vectors.	Economic load dispatch
[23]	It uses a chaotic based inertia weight strategy instead of fixed inertia weight or linearly decreasing inertia weight.	Allocation of power system stabilisers
[24]	This technique is a hybrid of PSO and bacterial foraging optimisation algorithm wherein the unit length random direction of tumble behaviour can be obtained by the global best position of PSO.	Coordinated design of power system stabiliser and static synchronous series compensator
[25]	PSO is hybridised with Nelder Mead algorithm in order to enhance its exploitative capability. Moreover, mutation operator is incorporated to enhance explorative capability.	Crack detection in cantilever beams
[26]	Chaotic initialisation is implemented instead of random initialisation. This chaotic-based PSO outperforms conventional PSO in lifetime optimisation of wireless networks.	Lifetime optimisation in wireless networks
[27]	PSO with fuzzy acceleration coefficients is hybridised with Nelder Mead algorithm to utilise its strong exploitative capability.	Optimal power flow in power systems
[28]	The concepts of imperialistic competitive algorithm (ICA) are adopted in PSO's flight equations. Moreover, crossover operator is incorporated.	Mathematical benchmarks, compression spring problem and welded beam problem.
[29]	Mutation and crossover operators are incorporated into PSO.	Curve fitting in manufacturing
[30]	PSO is hybridised with SA wherein PSO provides initial solutions for SA.	Time series forecasting

where C_1 and C_2 are called cognitive and social acceleration coefficients respectively, r_{1d} and r_{2d} are two random numbers in the interval [0,1].

However, primary PSO characterised by (1) and (2) did not work desirably [6]. Therefore, the inertia weight PSO was introduced wherein the velocity update equation is as follows [6]:

$$V_{id}(t+1) = \omega V_{id}(t) + C_1 r_{1d}(P_i - X_{id}) + C_2 r_{2d}(P_{gd} - X_{id}) \quad (3)$$

where ω is called inertia weight and is commonly decreased during the run.

After introducing primary versions of PSO, the researchers developed many different modified PSO variants to mitigate shortcomings of primary versions. The main focus in these modified variants is to enhance PSO's exploration capability and alleviate premature convergence problem [7–10]. In [11], incorporating a constriction factor was proposed to bound particles movements and enhance PSO's performance. In [12], PSO as an algorithm with strong exploitation capability and weak exploration capability is hybridised with artificial bee colony (ABC) as an algorithm with weak exploitation capability and strong exploration capability. In this hybridisation, the global best solutions obtained by the PSO and ABC are recombined and the solution obtained from this recombination is given to the populations of the PSO and ABC as the global best and neighbour food source for onlooker bees, respectively. Application of this hybrid algorithm to mathematical benchmark functions and energy demand estimation problem showed its superiority over ABC and PSO. Actually, the results indicated that the hybrid algorithm exposes strong capability both in exploration and exploitation.

In [3], Newton's laws of motion are incorporated into PSO. It exposes desirable convergence speed, solution accuracy and global optimality. In [13], in a variant called "compact PSO" the search logic of PSO is employed but instead of having an actual swarm of solutions, a probabilistic representation of the population is used. In [14] a new PSO variant is proposed wherein two strategies including a diversity enhancing mechanism and two neighbourhood search strategies are incorporated into PSO. The former strategy is helpful to increase the swarm diversity by adjusting the dissimilarities among particles. The latter strategy is beneficial for accelerating the convergence rate because of the attraction of the

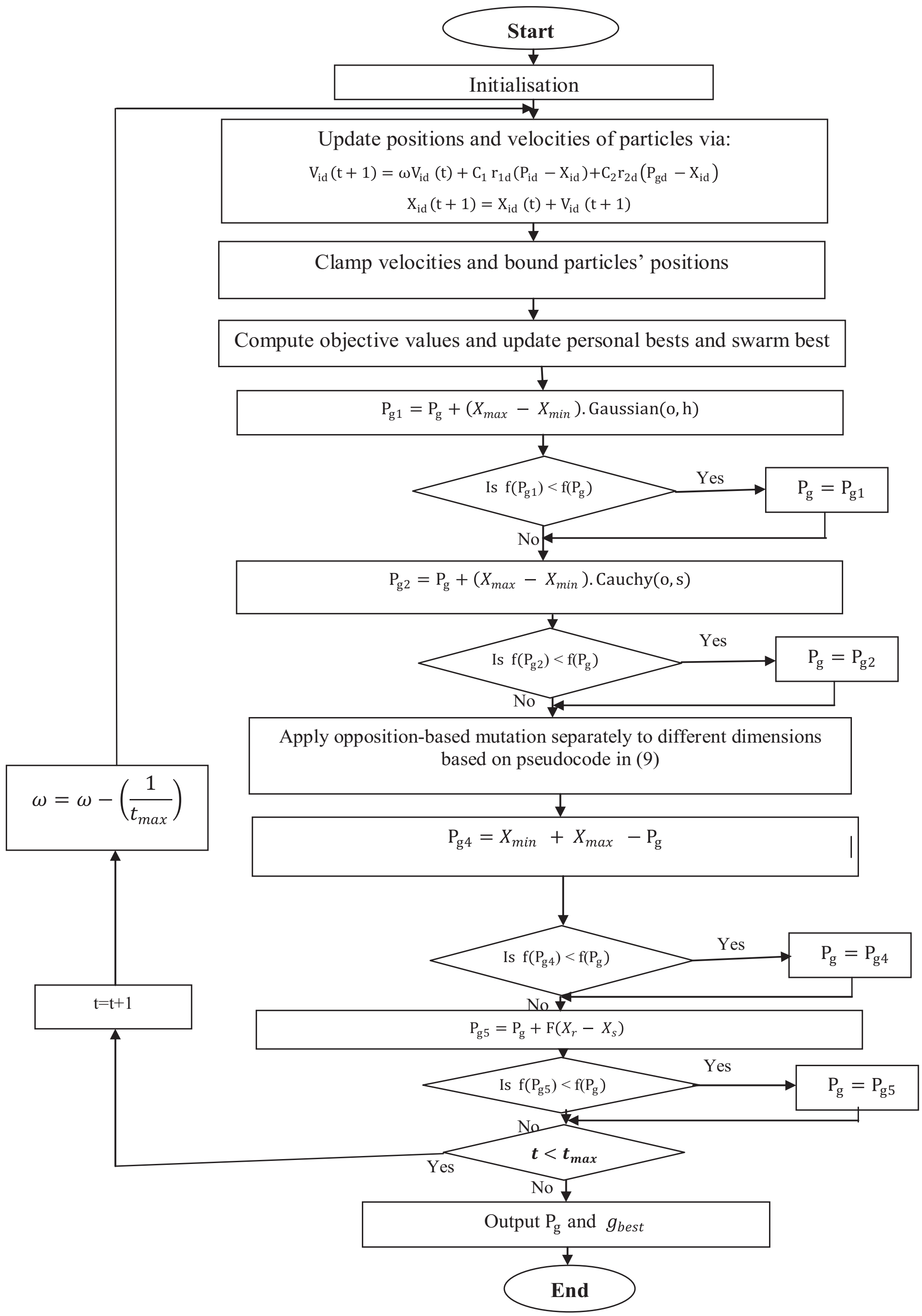

Fig. 1. Flowchart of ELPSO.

previous best particles and the global best particle. By combining these two strategies, an appropriate trade-off between exploration and exploitation is attained. In [15], cellular PSO is proposed which hybridises cellular automata (CA) and PSO. A review of some recent new PSO variants has been tabulated in Table 1.

3. Enhanced leader PSO

In conventional PSO, there exist two crucial issues; first issue is that since all particles are attracted towards swarm leader, they may converge prematurely without enough exploration of search space, that is, conventional PSO is prone to premature convergence. The latter issue is that if the particles are trapped in local optima, i.e., $P_i \cong X_i \cong P_g$, then according to Eq. (3), we have:

$$V_{id}(t+1) \cong \omega V_{id}(t) \tag{4}$$

Since $\omega < 1$, $V_{id} \to 0$, that is, the particles' velocities tend to be zero. Therefore, the particles cannot jump out from local optimum meaning that in conventional PSO, there is no mechanism for jumping out from local optima.

Table 2
Comparison of different algorithms for unimodal functions in $n = 5$ (mean of objective values is the comparison criterion).

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F1								
Mean	**3.20e−15**	2.176e−13	5.86e−5	3.95e−5	3.73e−7	4.53e−7	2.294e−4	5.28e−11
Std	**3.10e−15**	1.936e−13	1.031e−4	3.57e−5	3.09e−5	1.432e−6	1.265e−4	5.08e−11
Median	**2.02e−15**	1.55e−13	1.77e−5	2.90e−5	3.17e−5	0	2.239e−4	3.41e−11
Min	**1.5e−16**	3.43e−14	1e−7	2.1e−6	6e−6	0	3.42e−5	4.2e−12
Max	**1.145e−14**	9.984e−13	5.188e−4	1.648e−4	1.546e−6	4.528e−6	4.843e−4	1.780e−10
F2								
Mean	**5.17e−13**	1.683e−11	25.826	0.0179	0.0257	0.6330	0.0032	0.0023
Std	**5.88e−13**	1.892e−11	27.9079	0.0202	0.0633	0.1422	0.0015	0.0018
Median	**3.08e−13**	1.249e−11	19.5096	0.0116	0.0005	0.6688	0.0029	0.0017
Min	**5.3e−14**	1.7e−13	0.2531	0.0020	0	0.2563	0.0007	0.0001
Max	**2.669e−12**	9.100e−11	213.3639	0.1007	0.2631	0.7894	0.0066	0.0070
F3								
Mean	**1.644e−14**	9.47e−13	0.0358	2.653e−4	1.834e−6	2.0082	4.140e−4	0.0212
Std	**1.494e−14**	7.34e−13	0.0537	1.699e−4	1.453e−6	0.9323	1.920e−4	0.0225
Median	**1.281e−14**	6.15e−13	0.0109	2.474e−4	1.535e−6	2.0788	4.294e−4	0.0157
Min	**1.51e−15**	1.69e−13	0.0002	2.40e−5	2.15e−7	0.9748	1.531e−4	0.0008
Max	**7.225e−14**	2.703e−12	0.2192	6.471e−4	6.614e−6	3.7721	8.283e−4	0.1023

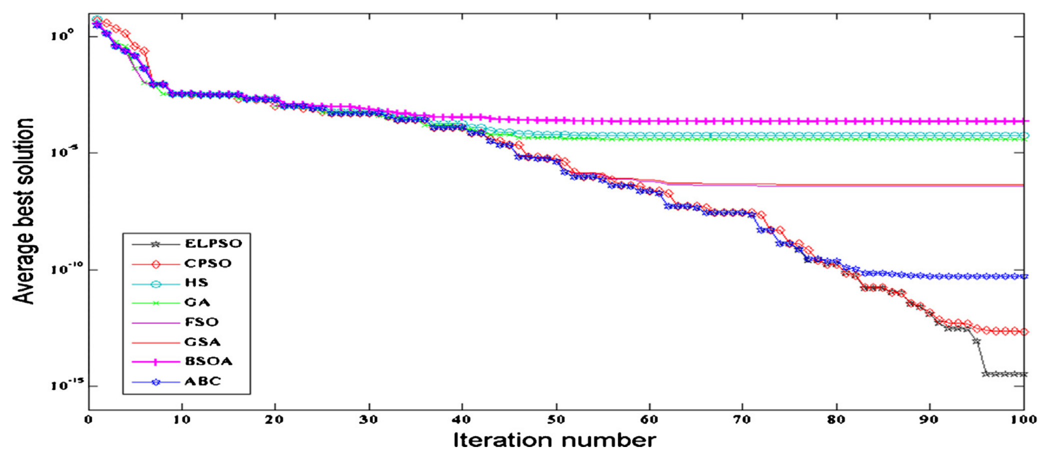

Fig. 2. Convergence curve of algorithms for F1 in $n = 5$.

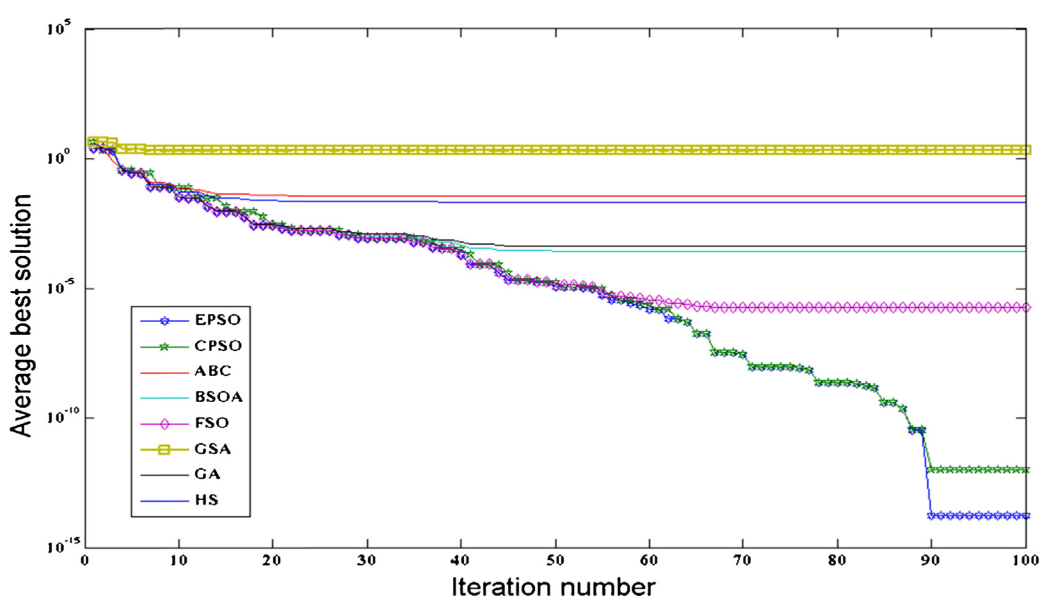

Fig. 3. Convergence curve of algorithms for F3 in $n = 5$.

Table 3
Comparison of different algorithms for multimodal functions in $n = 5$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F4								
Mean	0.4261	1.2939	0.0299	0.7679	1.9697	**0**	0.8239	4.46e−6
Std	0.1931	0.9111	0.0500	0.6894	1.2072	**0**	0.6577	5.53e−5
Median	0.3924	0.9950	0.0074	0.6071	1.9899	**0**	1.0256	2.41e−5
Min	0	0	0.0001	0.0523	0	**0**	0.0272	1.3e−7
Max	1.7849	3.9812	0.2243	3.0978	4.3670	**0**	2.0311	2.211e−5
F5								
Mean	**1.109e−13**	4.44e−13	0.0002	8.06e−5	2.52e−6	0.0081	9.70e−5	1.295e−10
Std	**1.103e−13**	4.62e−13	0.0005	4.92e−5	4.24e−6	0.0122	5.41e−5	8.96e−11
Median	**6.70e−14**	2.92e−13	0.0001	6.93e−5	7.9e−7	0.0033	8.44e−5	2.036e−10
Min	**1.16e−14**	2.5e−14	0	8.8e−6	1.1e−7	0.0005	2.75e−5	2.00e−10
Max	**4.165e−13**	2.349e−12	0.0025	1.834e−4	2.095e−5	0.0395	2.201e−4	3.535e−10
F6								
Mean	**0.0114**	0.0891	0.3356	0.1170	0.0415	7.5134	0.7219	0.0126
Std	**0.0039**	0.0411	0.1944	0.0421	0.0384	2.7942	0.5013	0.0050
Median	**0.0121**	0.0788	0.2939	0.1145	0.0292	7.4211	0.6283	0.0123
Min	**0.0010**	0.0271	0.0579	0.0490	0.0009	3.1910	0.0370	0.0010
Max	**0.0246**	0.1577	0.8597	0.2231	0.1555	12.9143	1.9514	0.0273
F7								
Mean	**0.0952**	0.1352	2.3439	1.0720	1.9456	32.6862	0.3366	0.1018
Std	**0.1988**	0.7172	2.0940	0.6049	1.1808	16.9822	0.2763	0.1241
Median	**0**	0	3.0453	1.0715	2.0208	29.2604	0.2597	0.0629
Min	**0**	0	0.0258	0.0483	0.0023	7.2135	0.0720	0.0039
Max	**0.9308**	3.9308	5.9952	2.1552	4.8742	64.1157	1.1010	0.5019
F8								
Mean	**1.62e−7**	3.012e−6	0.0888	0.0572	0.0046	0.0288	0.0293	2.799e−4
Std	**2.57e−8**	1.608e−6	0.0887	0.0245	0.0015	0.0347	0.0065	1.246e−4
Median	**1.30e−7**	2.349e−6	0.0552	0.0510	0.0049	0.0203	0.0298	2.731e−4
Min	**1.65e−8**	9.35e−7	0.0118	0.0224	0.0012	0	0.0191	4.65e−5
Max	**8.137e−7**	6.509e−6	0.4392	0.1150	0.0070	0.1051	0.0384	5.919e−4

In PSO, all particles are attracted towards swarm leader (P_g). So, having high quality leader can make the search process more efficient. Like a society or organisation wherein existence of a good leader can result in more success. Enhanced leader PSO (ELPSO) is a new PSO variant whose main characteristic is the enhancement of swarm leader at each iteration of search process [8]. In ELPSO, at each iteration, a five-staged successive mutation strategy is applied to swarm leader. After applying each mutation, if the mutated P_g has better fitness value than the current P_g, it takes the position of current P_g. By applying this successive mutation strategy to swarm leader, swarm leader is enhanced at each iteration, so a more efficient search process is achieved [8].

At the first stage of the successive mutation strategy, Gaussian mutation is applied to swarm leader as below.

$$P_{g1}(d) = P_g(d) + (X_{\max}(d) - X_{\min}(d)) \cdot \text{Gaussian}(o, h) \quad \text{for } d = 1, 2, \ldots, n \tag{5}$$

where $X_{\max}(d)$ and $X_{\min}(d)$ represent upper and lower bounds of decision vectors in dth dimension and h is standard deviation of Gaussian distribution. If the fitness of P_{g1} is better than fitness of P_g, then P_{g1} takes the position of P_g.

The standard deviation of the Gaussian distribution is decreased linearly during the run as Eq. (6). This is to ensure that the

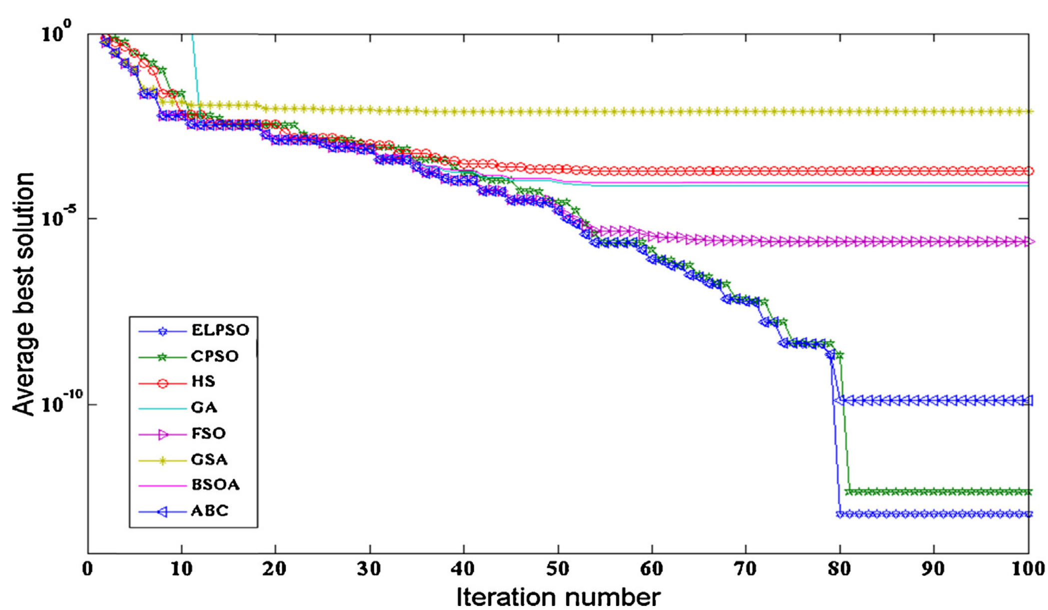

Fig. 4. Convergence curve of algorithms for F5 in $n = 5$.

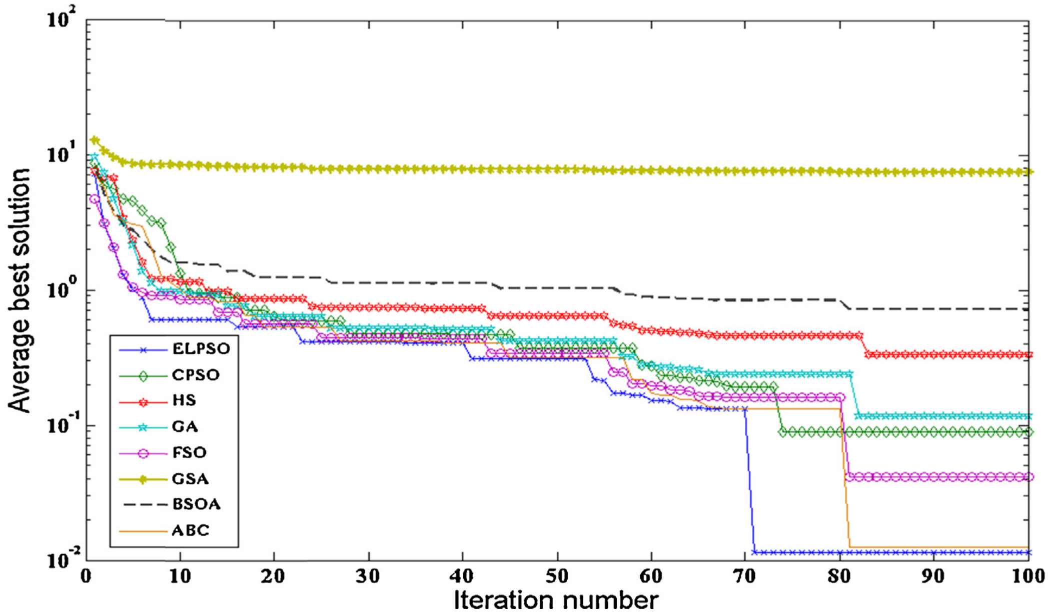

Fig. 5. Convergence curves for F6 in dimension 5.

exploration capability is stronger at initial iterations and it fades out during the run to have more exploitation capability.

$$h(t+1) = h(t) - \left(\frac{1}{t_{\max}}\right) \tag{6}$$

At the second stage of the successive mutation strategy, Cauchy mutation is applied to swarm leader as below.

$$P_{g2}(d) P_g(d) + (X_{\max}(d) - X_{\min}(d)) \cdot \mathrm{Cauchy}(o, s) \quad \text{for } d = 1, 2, \ldots, n \tag{7}$$

where s is scale parameter of Cauchy distribution which is decreased linearly during the run as Eq. (8). This is to ensure that the exploration capability is more at initial iterations and it fades out during the run to have more exploitation capability.

$$s(t+1) = s(t) - \left(\frac{1}{t_{\max}}\right) \tag{8}$$

If the fitness of P_{g2} is better than fitness of P_g, then P_{g2} takes the position of P_g.

At the third stage of the successive mutation strategy, opposition-based mutation is separately applied to all different dimensions of P_g as below.

$$P_{g3}(d) = X_{\min}(d) + X_{\max}(d) - P_g(d) \quad \text{for } d = 1, 2, \ldots, n \tag{9}$$

If the fitness of P_{g3} is better than fitness of P_g, then P_{g3} takes the position of P_g.

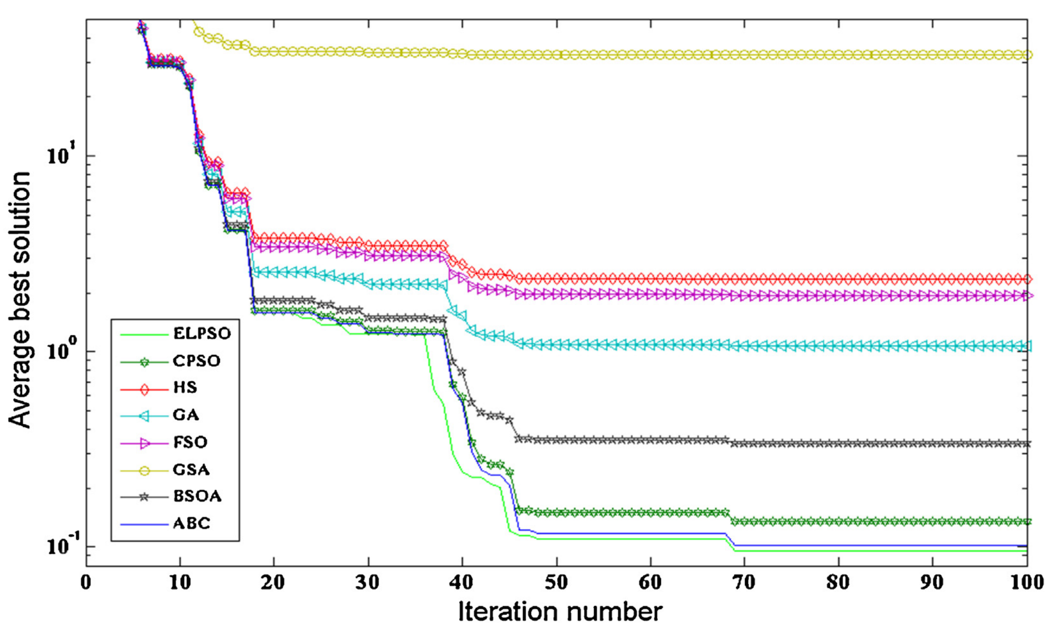

Fig. 6. Convergence curve of algorithms for F7 in $n = 5$.

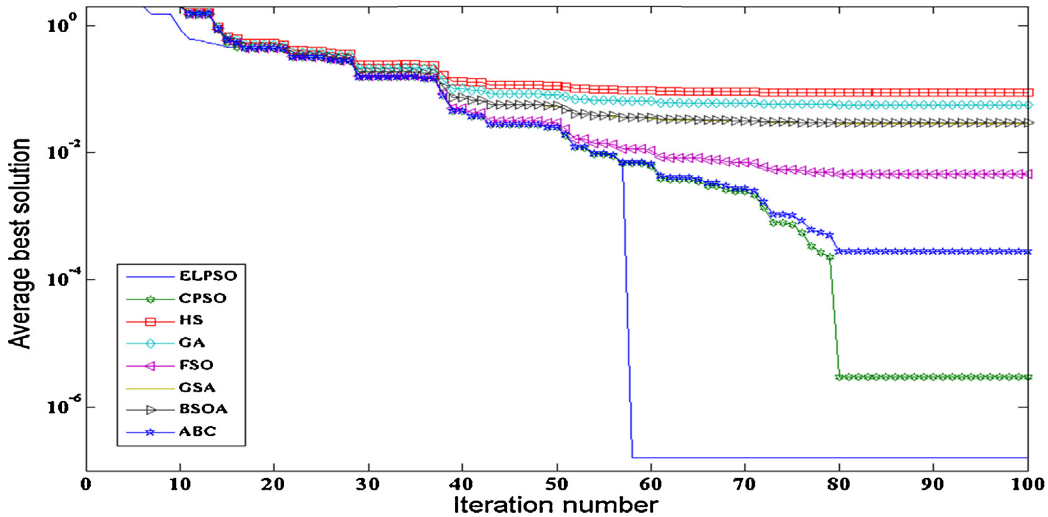

Fig. 7. Convergence curve of algorithms for F8 in $n = 5$.

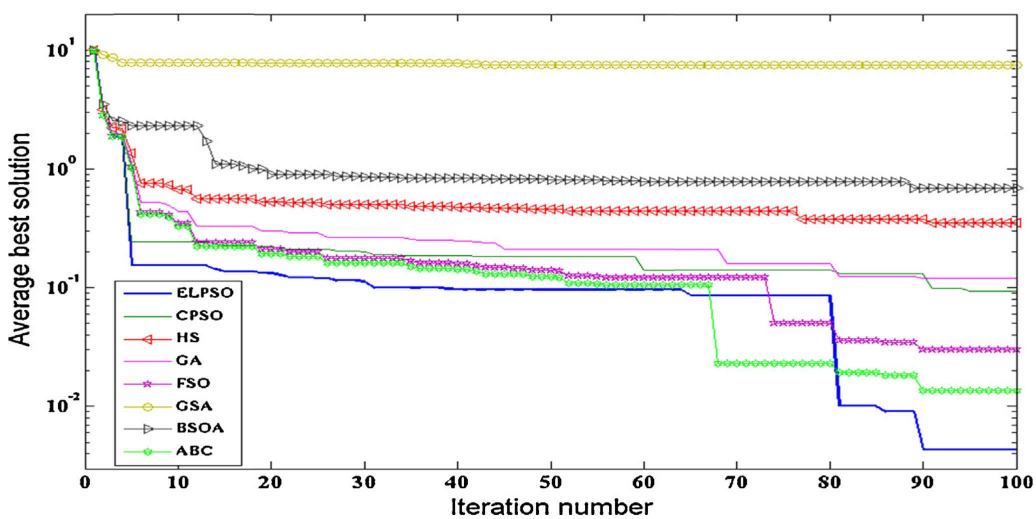

Fig. 8. Convergence curve of algorithms for rotated F6.

At fourth stage of the successive mutation strategy, opposition-based mutation is applied to the whole P_g as below.

$$P_{g4} = X_{\min} + X_{\max} - P_g \tag{10}$$

If the fitness of P_{g4} is better than fitness of P_g, then P_{g4} takes the position of P_g.

At fifth stage of the successive mutation strategy, DE-based mutation operator is applied to P_g as below.

Table 4
Comparison of different algorithms for rotated unimodal functions in $n = 5$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F1								
Mean	**5.03e−14**	2.49e−13	7.05e−5	4.615e−5	3.476e−7	1.17e−6	1.997e−4	8.48e−11
Std	**6.14e−14**	2.98e−13	1.271e−4	2.535e−5	2.209e−7	3.61e−6	9.09e−5	1.124e−10
Median	**2.71e−14**	1.84e−13	2.04e−5	4.052e−5	3.028e−7	0	1.881e−4	4.96e−11
Min	**3.9e−15**	1.3e−14	0	9.87e−6	9.56e−8	0	5.30e−5	5.1e−12
Max	**2.889e−13**	1.569e−12	6.258e−4	9.983e−5	8.729e−7	1.144e−5	3.542e−4	5.503e−10
F2								
Mean	**4.24e−12**	2.43e−11	18.6933	0.0863	0.0122	1.0667	0.0033	0.0019
Std	**3.49e−12**	2.53e−11	23.9449	0.2024	0.0171	0.3531	0.0014	0.0015
Median	**3.54e−12**	1.51e−11	10.3297	0.0115	0.0029	0.8862	0.0030	0.0015
Min	**2.0e−13**	1.9e−12	0.2609	0.0014	0.0000	0.7412	0.0014	0.0002
Max	**1.685e−11**	1.004e−10	95.1818	0.6669	0.0448	1.8222	0.0059	0.0059
F3								
Mean	**2.537e−13**	1.320e−12	0.0293	2.273e−4	1.626e−6	0.4551	0.0005	0.0200
Std	**2.187e−13**	1.907e−12	0.0368	1.432e−4	9.03e−7	0.3148	0.0002	0.0145
Median	**2.131e−13**	6.07e−13	0.0088	1.977e−4	1.651e−6	0.3994	0.0005	0.0160
Min	**1.75e−14**	4.4e−14	0.0003	1.83e−5	3.62e−7	0.0615	0.0001	0.0017
Max	**8.188e−13**	9.839e−12	0.1146	6.221e−4	3.064e−6	1.0793	0.0010	0.0571

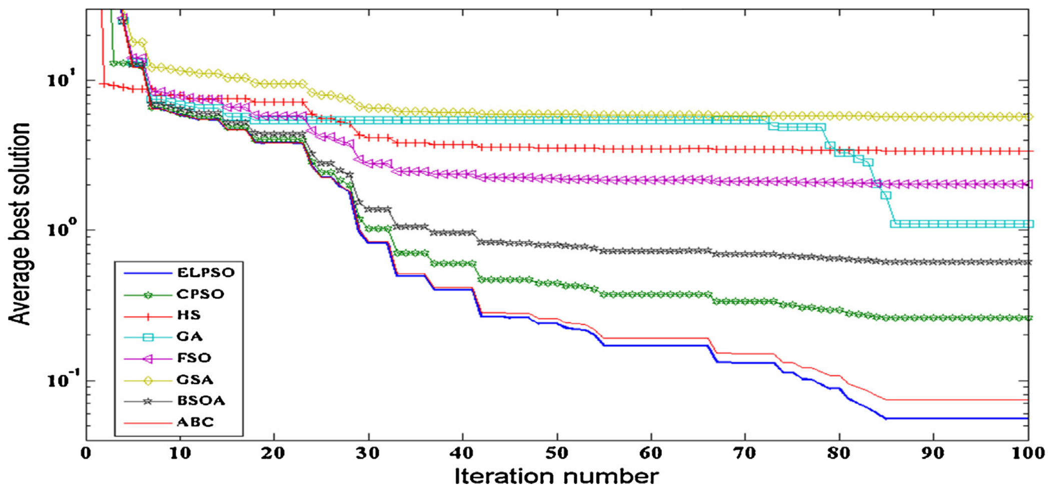

Fig. 9. Convergence curve of algorithms for rotated F7.

$$P_{g5} = P_g + F(X_r - X_s) \quad (11)$$

where r and s are two random unequal particles in swarm and F is a control parameter called scale factor.

If the P_{g5} is fitter than P_g, then P_{g5} takes the position of P_g.

ELPSO addresses both mentioned crucial issues of conventional PSO. Since at each iteration, different regions of search space are explored to find the best possible swarm leader, the premature convergence problem is expected to be mitigated. Moreover, the successive mutation strategy of ELPSO acts as a jumping out mechanism when the particles are trapped in local optima. The positive features of ELPSO are listed below [8].

1. In all five stages of successive mutations applied to P_g, P_g is mutated if the new P_g possesses better objective than the current P_g. In other words, the mutations are conditional. By conditional mutations used in ELPSO, P_g and the whole swarm are more attracted towards regions with good objective values and more quality solutions can be achieved.

Table 5
Comparison of different algorithms for rotated multimodal functions in $n = 5$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F4								
Mean	0.0141	1.9237	0.0215	1.6921	1.7910	0.3449	0.9752	**8.43e−6**
Std	0.0083	1.0430	0.0373	1.1018	0.9143	0.5652	0.6374	**8.43e−6**
Median	0.0199	1.9899	0.0051	1.5729	1.9900	0	1.0454	**4.84e−6**
Min	0	0	0	0.1346	0.0001	0	0.0231	**1.2e−7**
Max	0.0798	3.9798	0.1372	3.9168	2.9850	1.4171	2.1657	**3.169e−5**
F5								
Mean	**1.149e−13**	4.76e−13	0.0002	6.39e−5	2.127e−6	0.0278	9.24e−5	8.97e−11
Std	**1.187e−13**	7.62e−13	0.0003	4.92e−5	3.357e−6	0.0214	4.06e−5	9.55e−11
Median	**7.19e−14**	2.79e−13	0.0001	5.75e−5	7.20e−7	0.0272	8.39e−5	7.55e−11
Min	**6.2e−15**	1.2e−14	0	2.5e−6	1.69e−7	0.0004	4.01e−5	4e−13
Max	**4.431e−13**	4.231e−12	0.0010	2.453e−4	9.163e−6	0.0688	1.755e−4	5.115e−10
F6								
Mean	**0.0043**	0.0927	0.3492	0.1189	0.0300	7.4448	0.6866	0.0136
Std	**0.0014**	0.0589	0.2301	0.0405	0.0218	3.3767	0.4182	0.0065
Median	**0.0029**	0.0714	0.2938	0.1152	0.0229	6.7265	0.6432	0.0137
Min	**0.0007**	0.0105	0.0023	0.0441	0.0101	3.1479	0.1628	0.0003
Max	**0.034**	0.2587	0.7713	0.1921	0.0948	13.0600	1.5849	0.0292
F7								
Mean	**0.0551**	0.2621	3.3874	1.1055	2.0458	5.7634	0.6179	0.0746
Std	**0.0239**	0.9973	2.5074	0.6436	1.3827	1.7891	0.3179	0.0829
Median	**0**	0	3.5650	1.1506	1.8992	5.4878	0.6733	0.0440
Min	**0**	0	0.0357	0.0373	0.1552	2.9368	0.0928	0.0081
Max	**0.1308**	3.9308	11.5375	2.6914	4.5985	9.4227	1.2382	0.3477
F8								
Mean	**4.65e−7**	3.35e−6	0.0433	0.0630	0.0044	0.0924	0.0273	2.058e−4
Std	**2.38e−7**	2.60e−6	0.0604	0.0216	0.0014	0.0915	0.0087	7.82e−5
Median	**4.04e−7**	2.50e−6	0.0206	0.0562	0.0045	0.0709	0.0274	2.045e−4
Min	**1.50e−7**	6.4e−7	0.0014	0.0353	0.0014	0	0.0108	3.40e−5
Max	**1.219e−6**	1.331e−5	0.2535	0.0897	0.0080	0.2680	0.0470	3.583e−4

Table 6
Wilcoxon signed-rank test for dimension 5.

ELPSO vs	CPSO	HS	GA	FSO	GSA	BSOA	ABC
Dimension 5	4.3778e−4	0.0038	4.3778e−4	4.3778e−4	0.0019	4.3778e−4	0.0386

Table 7
Comparison of different algorithms for unimodal functions in $n = 30$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F1								
Mean	**5.244e−8**	6.11e−8	1.0389	2.0586	1.819e−5	0.0237	0.0419	0.4114
Std	**1.643e−8**	3.64e−8	0.5401	0.5491	4.92e−6	0.0019	0.0051	0.2810
Median	**5.176e−8**	5.92e−8	0.8989	2.0860	1.721e−5	0.0230	0.0416	0.3935
Min	**2.979e−8**	1.36e−8	0.5550	1.0484	9.79e−6	0.0221	0.0321	0.0265
Max	**7.614e−8**	1.500e−7	1.9489	2.9578	3.303e−5	0.0259	0.0500	1.2233
F2								
Mean	**0.6174**	0.6919	463.98	347.3291	1.0371	4.0109	1.5656	44.0769
Std	**0.1204**	0.0691	72.97	119.9879	0.9097	0.9344	0.4185	32.0594
Median	**0.5667**	0.6689	461.43	357.0357	0.7326	3.7168	1.4597	33.9093
Min	**0.2169**	0.6667	387.12	123.5378	0.6679	3.2589	1.1742	7.9918
Max	**0.9504**	0.9988	568.26	594.5779	5.4393	5.0570	2.3224	115.3589
F3								
Mean	**0.0003**	0.0005	294.6300	54.0905	0.9660	58.16	0.1540	242.2357
Std	**0.0003**	0.0003	58.4089	10.2974	0.9866	29.73	0.0223	31.8634
Median	**0.0002**	0.0004	298.0227	53.4953	0.6918	43.17	0.1527	249.7921
Min	**0.0001**	0.0001	210.4752	30.7910	0.0094	21.41	0.1215	183.5093
Max	**0.001**	0.0015	368.5845	72.7005	4.0420	173.32	0.1983	292.7280

Table 8
Comparison of different algorithms for multimodal functions in $n = 30$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F4								
Mean	**8.6403**	25.6700	24.3984	142.0224	29.7863	5.4160	42.7192	62.8410
Std	**4.1871**	5.3637	21.9022	15.2892	5.8596	3.1959	6.3649	7.3062
Median	**8.8185**	26.8639	18.7426	139.5265	29.8518	4.9023	43.3136	64.5771
Min	**3.8941**	13.9294	9.3234	109.9212	18.9075	2.5080	29.7170	52.4758
Max	**18.8062**	33.8287	62.5372	171.2527	39.8013	8.8376	51.9668	77.3247
F5								
Mean	0.1986	0.4689	0.2970	2.3619	**0.0031**	1.8517	1.1454	0.7010
Std	0.0717	0.4229	0.1891	0.4511	**0.0163**	0.1468	1.6001	0.5540
Median	0.1087	0.3581	0.2575	2.2331	**0.0001**	1.9277	0.3035	0.5996
Min	0	0	0.0521	1.5838	**0**	1.6825	0.0186	0.0801
Max	0.5420	1.6360	0.5341	3.4032	**0.0896**	1.9449	4.5853	2.0266
F6								
Mean	**2.748e−4**	0.0135	6.0126	7.9564	0.0083	305.1743	15.9278	1.9129
Std	**1.232e−4**	0.0130	0.2969	1.4340	0.0020	5.0033	3.8231	1.0371
Median	**2.547e−4**	0.0089	5.9205	7.6865	0.0082	307.8109	15.7031	1.5246
Min	**1.628e−4**	0.0002	5.6253	5.3898	0.0054	299.4042	11.1064	1.1524
Max	**4.068e−4**	0.0520	6.3507	11.8630	0.0137	308.3080	22.8020	5.2249
F7								
Mean	**5.8172**	52.7099	1452.9	728.1	58.7722	1414.8	51.2181	283.8412
Std	**1.3840**	31.4375	472.7	193.8	34.2257	287.0	33.3722	113.1524
Median	**5.8938**	29.2173	1452.5	731.9	29.3760	1509.4	32.6838	258.9218
Min	**1.1337**	4.4676	887.2	432.3	24.8685	1005.6	29.2770	143.4207
Max	**9.4242**	110.5213	1963.2	1307.2	104.6797	1634.8	111.4459	534.3106
F8								
Mean	0.2779	1.8896	4.8483	5.8373	**0.0141**	13.8286	0.2151	8.0401
Std	0.0816	0.5905	0.6852	0.3351	**0.0020**	0.1833	0.0188	0.7006
Median	0.3169	2.0099	4.8105	5.8533	**0.0138**	13.7784	0.2192	8.0698
Min	0.0780	0.0009	3.6992	5.3578	**0.0093**	13.5839	0.1737	6.4486
Max	0.7389	2.8141	6.0914	6.5378	**0.0173**	14.1255	0.2397	9.2224

2. By using five successive mutations, different regions of search space are searched efficiently, therefore premature convergence probability decreases. Actually, at each iteration, in stage 1 (Gaussian mutation) one, in stage 2 (Cauchy mutation) one, in stage 3 (opposition-based mutation on dimensions) "n", in stage 4 (opposition-based mutation on P_g as a whole) one and in stage 5 (DE-based mutation) one new position for finding better leader is investigated. Therefore, at each iteration, "$n+4$" new positions for finding better swarm leader are tested. The position with the best objective value among these "$n+4$" cases and the position achieved by PSO flight equations is determined as the swarm leader for that iteration. This feature diminishes premature convergence probability. For better imagination, assume an optimisation problem whose dimension is 20. While in conventional PSO, at each iteration one position is swarm best, in ELPSO, at each iteration 25 positions are tested and the best one is determined as swarm leader, so we have an enhanced swarm leader. However, in ELPSO, number of function evaluations for the same number of iterations and particles is different from number of function evaluations in conventional PSO. That is, unlike

Table 9
Wilcoxon test results for dimension 30.

ELPSO vs	CPSO	HS	GA	FSO	GSA	BSOA	ABC
Dimension 30	0.0078	0.0078	0.0078	0.1484	0.0391	0.0234	0.0078

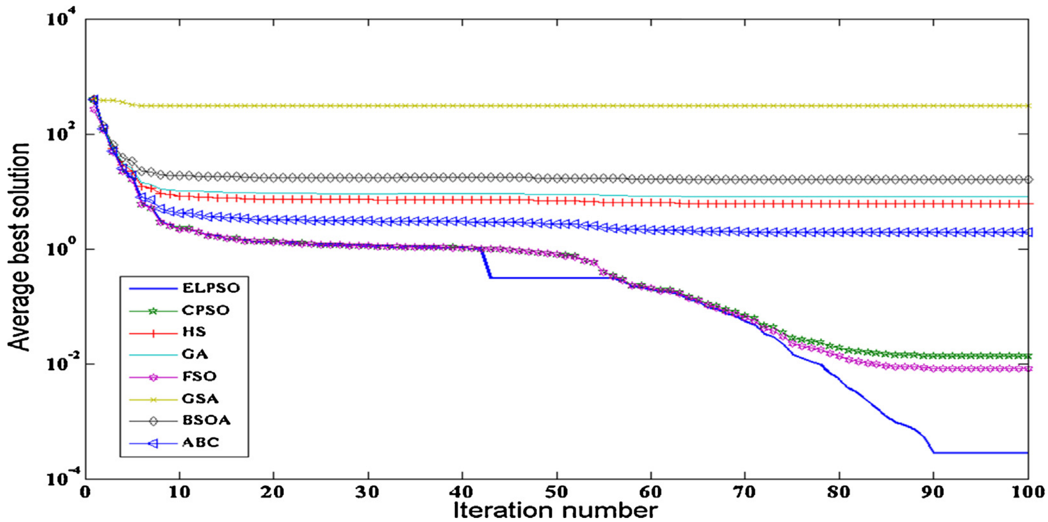

Fig. 10. Convergence curves for F6 in dimension 30.

conventional PSO whose number of function evaluations is computed by Eq. (12), in ELPSO, it is computed by Eq. (13).

$$NFE_{CPSO} = N_p \cdot (1 + t_{max}) \tag{12}$$

$$NFE_{ELPSO} = N_p \cdot (1 + t_{max}) + (n + 4) t_{max} \tag{13}$$

where NFE_{CPSO} and NFE_{ELPSO} represent number of function evaluations in conventional PSO and ELPSO respectively, N_p is number of particles and t_{max} is maximum number of iterations.

3. In ELPSO, explorative capability decreases during the run, because in stages 1 and 2 of successive mutation (Gaussian and Cauchy mutations), the standard deviation and scaling parameter are decreased during the run. This property is very important for optimisation algorithms.
4. Unlike most existing mutation operators which are applied to positions and velocities of particles, in ELPSO mutations are applied to swarm leader. Therefore, the leader is improved which attracts all particles towards better regions of search space.
5. Mutations can be classified into two groups; mutations with short jump and mutations with long jump. The mutations with short jump can reduce premature convergence probability in PSO but are not able to jump out particles from local optimum when the swarm stagnates. On the other hand, mutations with long jump are efficient jump-out mechanisms and can successfully jump out particles from local optimum after occurrence of stagnation. In ELPSO, stages 1, 2 and specially 3 are considered long jump mutations whose presence may make the PSO very capable in jumping out local optimum after stagnation.

The flowchart of ELPSO is illustrated in Fig 1 [8]. Its pseudocode has also been presented in Appendix A.1.

4. Experimental results

In this section, the proposed ELPSO is validated by comparing with some well-known optimisation algorithms. ELPSO is compared with conventional PSO (CPSO) [6], harmony search (HS) [31], genetic algorithm (GA) [32], firefly swarm optimisation (FSO) [33], gravitational search algorithm (GSA) [34], brainstorm optimisation (BSOA) [35] and artificial bee colony (ABC) [36]. Moreover, ELPSO will be compared with some modified variants of PSO and differential evolution (DE) including opposition-based DE (ODE) [37,38], competitive DE (CDE) [39], fixed inertia weight PSO (FIW-PSO), chaotic inertia weight PSO, random inertia weight PSO, time varying acceleration coefficient PSO (TVACPSO) and constricted

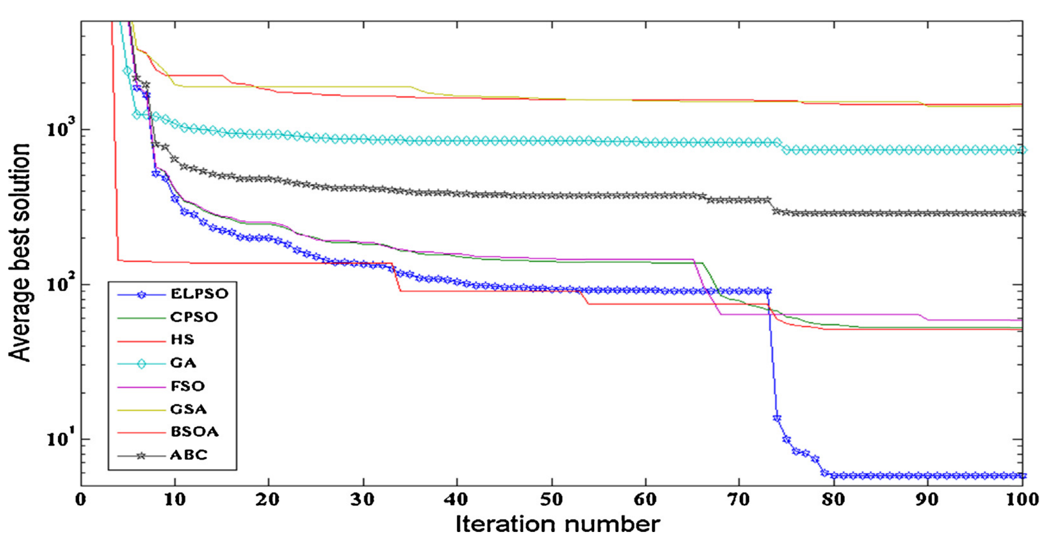

Fig. 11. Convergence curves for F7 in dimension 30.

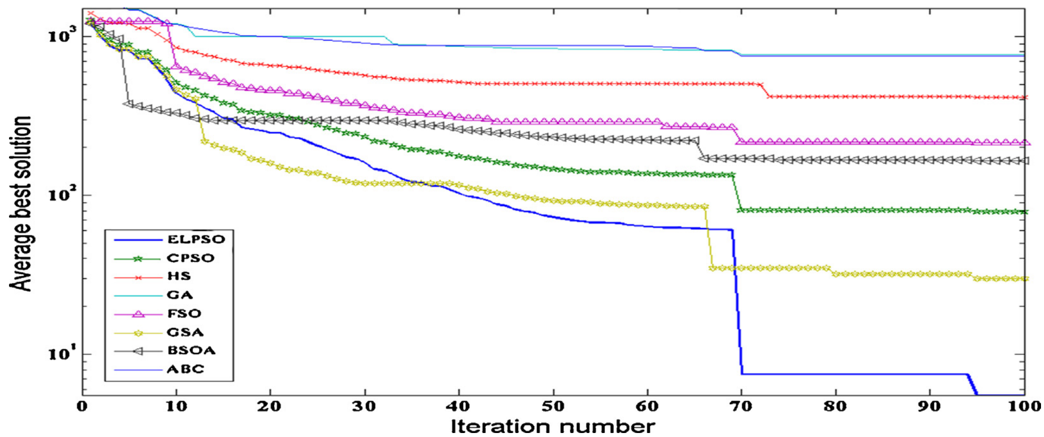

Fig. 12. Convergence curves for F4 in dimension 100.

PSO. Maximum number of iterations is set to 100 and number of individuals is set 50 times the number of decision variables. All algorithms are compared with the same number of function evaluations [40–44]. Since in ELPSO due to successive mutations, at each iteration more number of function evaluations is required with respect to other algorithms, maximum number of iterations in ELPSO is chosen less than conventional PSO and other algorithms. Indeed, for each problem dimension, maximum number of iterations in ELPSO is computed using Eqs. (12) and (13). All experiments are conducted for 30 independent runs and their statistical data are presented. The specifications of benchmark test functions are presented in Table A.1. In all benchmark functions the true global optimum is equal to zero. The experiments are separately conducted on low dimension functions, medium dimension functions, high dimension functions and rotated functions to fully evaluate the performance of ELPSO in different situations. It should be mentioned that in Sections 4.1–4.5, the statistical data of achieved optimal objectives for a certain number of function evaluations are recorded while in Section 4.7, the average of required number of function evaluations to achieve a certain quality in objective values is recorded. Wilcoxon signed-rank test is conducted to statistically evaluate the achieved results. It is done in a pair-wise way.

For CPSO and ELPSO, $C_1 = C_2 = 2$ and ω is linearly decreased from 0.9 to 0.4. For ELPSO, standard deviation of Gaussian mutation is 1, scale parameter of Cauchy mutation is 2 and scale factor of DE-based mutation is 1.2. For HS, harmony memory consideration rate is 0.9, initial and final values of pitch adjustment rate are 0.9 and 0.3 respectively, initial and final values of bandwidth are 0.5 and 0.2 respectively. For GA, crossover rate is 0.8 and number of elite individuals is 2. For FSOA, randomness factor is 0.5, attraction coefficient is 0.6 and attraction exponent is 1. In GSA, the power of denominator of force equation is set 1, coefficient and exponent of gravitational constant equation are set as 100 and 25 respectively. In BSOA, number of clusters is 5, the probability of selection of a cluster for creating new individual is 0.8 and the probability of replacement of a cluster centre by a random centre is set as 0.2. In ABC, “limit” as the sole control parameter is set as 50. For opposition-based DE (ODE), scaling factor is 0.5, crossover probability constant is 0.9, jumping rate constant is 0.3 and mutation strategy is DE/rand/1/bin.

4.1. Results for problems with low dimension (dimension 5)

In this dimension, number of individuals is set to 250, maximum number of iterations for all algorithms except ELPSO is 100. For fair comparison, in ELPSO, using Eqs. (12) and (13), maximum number of iterations is computed as 96. This is to ensure that all algorithms are compared with the same number of function evaluations. The statistical data of achieved objective values are tabulated in Tables 2 and 3. The best value of each row of a table is bolded. Indeed, Table 2 represents the results on unimodal functions and Table 3 represents the results on multimodal functions. Mean of objective values is the criterion of comparison among algorithms. Table 2 approves the strong exploitation capability of both conventional PSO and ELPSO, since the performance of an optimisation

Table 10
Comparison of different algorithms for unimodal functions in $n = 100$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F1								
Mean	**6.035e-5**	5.398e−4	76.0844	29.4081	2.211e−4	0.0672	0.3696	230.8250
Std	**1.860e−5**	2.639e−4	6.3151	4.6242	2.03e−5	0.0102	0.0222	19.3176
Median	**5.811e−5**	4.095e−4	78.0859	31.2806	2.216e−4	0.0691	0.3720	237.4058
Min	**4.297e−5**	2.708e−4	69.0111	23.7734	2.001e−4	0.0562	0.3406	198.6946
Max	**7.997e−5**	9.951e−4	81.1562	35.0934	2.524e−4	0.0763	0.3937	249.1387
F2								
Mean	0.7274	9.4802	1534.4	935.6	6.9207	**0.6667**	30.9772	490.83
Std	0.1633	5.6173	132.3	115.6	7.3965	**0**	9.5781	119.39
Median	0.6986	7.1689	1601.4	925.2	2.3565	**0.6667**	29.4125	454.80
Min	0.1047	3.2035	1382.1	768.7	1.1001	**0.6667**	21.2815	372.56
Max	1.2788	19.9524	1619.8	1045.4	17.7046	**0.6667**	43.8021	684.09
F3								
Mean	**2.5016**	40.9341	2694.91	412.8078	74.4212	171.3504	153.6407	1278.7
Std	**0.9689**	17.8104	462.561	53.3866	7.4873	5.8855	61.7728	129.1
Median	**2.2814**	39.9646	488.3287	382.7739	75.6089	168.1632	130.7713	1337.6
Min	**0.2268**	17.7263	287.5234	369.4271	62.8325	167.7459	91.0035	1052.2
Max	**6.9966**	78.0371	803.4160	476.6564	81.3081	178.1421	253.3970	1356.4

Table 11
Comparison of different algorithms for multimodal functions in $n = 100$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F4								
Mean	**5.5402**	78.7820	415.5759	768.2887	214.1549	29.8488	164.6546	753.3054
Std	**1.1612**	16.5476	171.3781	29.2261	30.1920	1.9899	44.1605	12.2647
Median	**5.7829**	82.1870	360.5646	775.2940	218.9371	29.8488	146.3873	756.6776
Min	**1.0448**	49.8618	300.2257	720.2649	169.1787	27.8589	135.4258	735.1485
Max	**9.7930**	95.7210	713.7281	797.1519	246.7955	31.8387	230.4178	767.0975
F5								
Mean	**0.0979**	2.2629	88.0076	39.8704	0.1629	3.920	15.5562	220.2077
Std	**0.0414**	1.1824	7.5948	2.0191	0.1632	0.274	6.0828	39.4847
Median	**0.0730**	1.9888	87.5581	40.9852	0.0900	3.872	14.9224	221.0249
Min	**0.0352**	1.1975	79.3889	37.2367	0.0899	3.673	8.8127	158.5157
Max	**0.7355**	5.3615	97.5253	41.7353	0.4548	4.215	23.5673	267.4462
F6								
Mean	0.2612	0.2841	271.3360	104.8670	**0.0152**	1.1321	83.5311	817.3863
Std	0.1092	0.0410	28.4434	11.0502	**0.0011**	0.3453	18.4563	63.1554
Median	0.2127	0.2793	268.3095	108.7177	**0.0151**	0.6543	84.5294	844.5736
Min	0.1846	0.2347	244.5269	86.0552	**0.0136**	0.2126	60.9781	722.3432
Max	0.3863	0.3603	301.1716	113.7519	**0.0165**	1.4532	104.0877	879.0898
F7								
Mean	**8.7390**	209.9321	3.1929e5	2.6131e4	97.8001	93.0768	206.5988	1.3808e6
Std	**2.5665**	83.3498	2.620 e4	3.856 e3	0.9119	0.1221	35.1550	1.919e5
Median	**8.3733**	225.3372	3.1715 e5	2.5822 e4	97.7481	93.0122	215.7586	1.2783e6
Min	**3.7441**	70.9597	2.9424e5	2.1330 e4	96.4872	93.0005	156.3850	1.2527e6
Max	**18.0996**	306.3208	3.4649 e5	3.1699 e4	98.7111	93.2176	238.4931	1.7100e6
F8								
Mean	0.2255	4.7431	13.3999	10.7037	**0.0275**	11.4327	8.0122	17.5914
Std	0.0867	0.4968	0.4734	0.4255	**0.0014**	2.6435	1.3962	0.2640
Median	0.1589	4.6822	13.5271	10.5791	**0.0273**	6.0956	7.9955	17.5324
Min	0.0475	3.6588	12.8760	10.2225	**0.0256**	1.7456	6.4311	17.1928
Max	0.8427	5.5736	13.7967	11.2436	**0.0293**	15.8561	9.6269	18.0125

algorithm for solving unimodal problems merely depends on its exploitation capability. The table shows that in solving unimodal problems in dimension 5, ELPSO surpasses CPSO and all other compared optimisation algorithms. After ELPSO, CPSO exposes the best performance.

Table 3 shows that for Levy, Griewank, Rosenbrock and Ackley functions, ELPSO outperforms all other optimisation algorithms. The only exception is Rastrigin wherein GSA outperforms ELPSO. The results in this table show that the successive mutation strategy of ELPSO leads to more efficient exploration of search space and significantly mitigates premature convergence problem. Figs. 2–7 show the convergence curves of different algorithms for some benchmark functions. Indeed, Figs. 2 and 3 illustrate strong exploitation capability of ELPSO and Figs. 4–7 indicate strong exploration capability of ELPSO.

4.2. Results for rotated functions in dimension 5

Here, number of individuals and maximum number of iterations is set the same with Section 4.1. The statistics of applying ELPSO and other optimisation algorithms to rotated functions are tabulated in Tables 4 and 5. Table 4 represents the results on rotated unimodal functions and Table 5 represents the results on rotated multimodal functions. Table 4 shows excellent performance of ELPSO for solving rotated unimodal functions. Indeed, in all used functions, ELPSO outperforms all other algorithms. Table 5 indicates that for rotated F5, rotated F6, rotated F7 and rotated F8, ELPSO outperforms all other algorithms. Just for rotated F4, ABC exposes better performance than ELPSO. Figs. 8 and 9 illustrate the convergence curves for rotated F6 and rotated F7. The curves testify strong exploration capability of ELPSO.

Now, Wilcoxon signed-rank test is conducted to statistically validate the outperformance of ELPSO over other algorithms in dimension 5. It is done in a pair-wise way. Table 6 shows p-values obtained by Wilcoxon signed-rank test for results in dimension 5. The p-values are very small which indicate the significant outperformance of ELPSO over all other compared algorithms.

4.3. Results for problems with medium dimension (dimension 30)

In this dimension, number of individuals is set to 1500, maximum number of iterations for all algorithms except ELPSO is 100. For fair comparison, in ELPSO, using Eqs. (12) and (13), maximum number of iterations is computed as 97. Tables 7 and 8 tabulate the results of different algorithms for dimension 30 in unimodal and multimodal functions respectively. Table 7 shows that for all unimodal problems (F1, F2 and F3), ELPSO exposes the best performance. Table 8 indicates that for F4, F6 and F7, ELPSO exposes the

Table 12
Wilcoxon test results for dimension 100.

ELPSO vs	CPSO	HS	GA	FSO	GSA	BSOA	ABC
Dimension 100	0.0078	0.0078	0.0078	0.1484	0.0078	0.0078	0.0078

Table 13
Wilcoxon results considering all dimensions.

ELPSO vs	CPSO	HS	GA	FSO	GSA	BSOA	ABC
	7.9529e−7	2.6866e−6	7.9529e−7	3.8116e−4	2.0138e−5	2.4514e−6	1.1116e−5

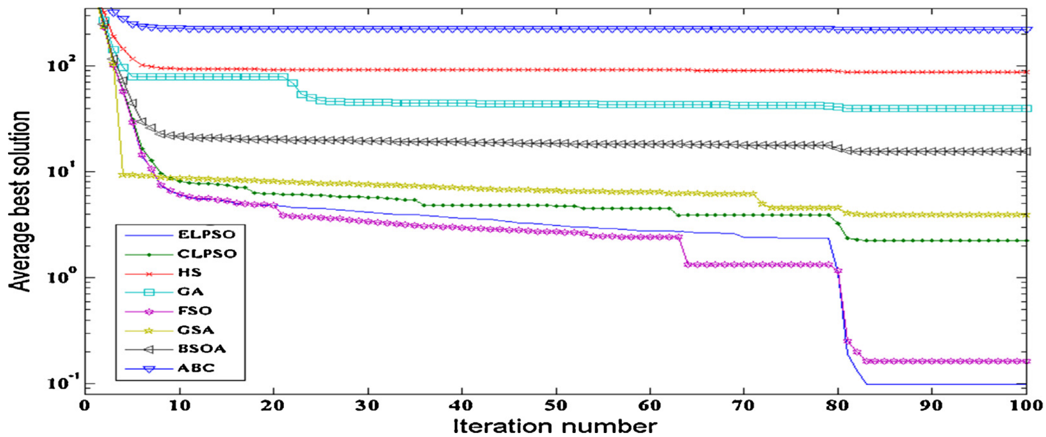

Fig. 13. Convergence curves for F5 in dimension 100.

Table 14
The average number of function evaluations to reach a required accuracy level in $n=5$.

	ELPSO	CPSO	HS	GA	FSO	GSA	BSOA	ABC
F1	35,992	**25,825**	363,625	95,124	4.5412e5	19,420	277,512	26,366
F2	72,094	**37,008**	Inf	Inf	Inf	Inf	281,518	209,620
F3	**27,512**	30,725	447,120	168,140	Inf	192,526	278,428	280,708
F4	44,387	3.8133e5	2.5133e5	485,464	Inf	Inf	2.6767e5	**25,904**
F5	**5856**	80,077	63,011	158,649	259,606	60,421	8250	1.1467e4
F6	209,863	Inf	Inf	Inf	Inf	Inf	Inf	**54,066**
F7	**29,757**	3.1008e4	306,150	479,357	Inf	220,621	Inf	218,309
F8	**22,519**	25,075	278,452	475,308	441,086	136,921	268,252	30,305
Rotated F1	45,201	2.4983e4	382,607	101,562	455,059	180,204	2.7658e5	**7.2334e3**
Rotated F2	**26,752**	37,775	Inf	Inf	Inf	Inf	284,753	27,634
Rotated F3	**28,903**	3.0417e4	459,257	175,387	4.9367e5	192,448	278,125	57,809
Rotated F4	175,356	Inf	263,602	483,784	Inf	Inf	2.6683e5	**26,118**
Rotated F5	**9503**	81,208	4751	161,276	2.6467e5	49,900	7.4167e3	10,106
Rotated F6	125,679	Inf	Inf	Inf	1.4033e5	Inf	Inf	**55,866**
Rotated F7	**75,507**	4.6433e5	333,908	435,491	Inf	219,039	Inf	116,663
Rotated F8	**21,751**	25,302	288,056	478,458	4.3967e5	143,030	2.6833e5	30,264

best performance while for F5 and F8, FSO provides the best results. Figs. 10 and 11 show convergence curves of F6 and F7 in dimension 30. They show strong exploration capability of ELPSO. Especially, see Fig. 11 wherein ELPSO is capable of finding near global solution while other algorithms provide very bad results and are trapped in local valleys. Table 9 tabulates the p-values achieved by Wilcoxon signed-rank test for results of dimension 30. The p-values indicate that ELPSO significantly outperforms all other algorithms except FSO. Although, in this dimension, ELPSO outperforms FSO, its p-value is not small enough.

4.4. Results for problems with high dimension (dimension 100)

In this dimension, number of individuals is set as 5000, maximum number of iterations for all algorithms except ELPSO is 100. For fair comparison, in ELPSO, using Eqs. (12) and (13), maximum number of iterations is computed as 98. Tables 10 and 11 illustrate that that for F1, F3, F4, F5 and F7, ELPSO's performance is superior to all other optimisation algorithms. Only in F2 and F6, GSA outperforms ELPSO. The tables indicate that both in unimodal and multimodal problems, performance of ELPSO is not highly affected by increase in dimensionality. In other words, ELPSO has proved to be scalable. Figs. 12 and 13 illustrate convergence curves of different algorithms in dimension 100. Table 12 tabulates the p-values achieved by Wilcoxon signed-rank test for results of dimension 100. The p-values indicate that in this dimension, like dimension 30, ELPSO significantly outperforms all algorithms except FSO. Although, in this dimension, ELPSO performs better than FSO, its p-value is not small enough.

4.5. Comparison of ELPSO with other PSO variants

In this section, the performance of ELPSO is compared with some modified variants of PSO and DE including linearly deceasing inertia weight PSO (LDIWPSO), fixed inertia weight PSO (FIWPSO), chaotic inertia weight PSO (CIWPSO), random inertia weight PSO (RIWPSO), time varying acceleration coefficient PSO (TVACPSO), constricted PSO (COPSO), opposition-based DE (ODE) and competitive DE (CDE). The results for dimensions 5 and 30 are provided in Appendix A.3. Tables A.2–A.5 show overall outperformance of ELPSO over other compared PSO variants for dimension 5 and dimension 30. Table A.6 shows p-values of ELPSO versus modified variants of PSO and DE, obtained by Wilcoxon signed-rank test. The p-values are small and strongly approve the significant outperformance of ELPSO over modified variants of PSO and DE.

4.6. Overall comparison of all algorithms

In this section, all algorithms are compared based on their overall results (considering all tested dimensions). Table 13 shows p-values obtained by Wilcoxon signed-rank test, considering the results of all dimensions. The p-values are very small and strongly approve the significant outperformance of ELPSO over other algorithms.

4.7. Convergence rate analysis

In this section, convergence rate of ELPSO and other used algorithms are evaluated. To this end, the average number of function evaluations required to achieve a certain fitness quality is computed for all algorithms. Each algorithm is run for 30 times. For unimodal functions, the required quality is 1e−6 whereas for multimodal ones, it is 0.01. Table 14 shows the average number of function evaluations for different algorithms. If an algorithm fails to achieve the required quality after 1e5 function evaluations, its corresponding cell in the table is set as infinite. According to the Table 14, in most functions (F3, F5, F7, F8, rotated F2, rotated F3, rotated F5, rotated F7 and rotated F8), ELPSO shows the best convergence rate among all compared algorithms. For F1 and F2, CPSO shows the best convergence rate. For functions F4, F6, rotated F1, rotated F4 and rotated F6, ABC offers the best convergence rate.

5. Conclusions

In this paper, a novel optimisation algorithm, named as enhanced leader PSO (ELPSO), has been introduced to mitigate premature convergence problem of conventional PSO. ELPSO is mainly based on a five-staged successive mutation strategy which is applied to swarm leader at each iteration. For validating ELPSO, its performance is compared with linearly decreasing inertia weight PSO (CPSO), opposition-based differential evolution (ODE), competitive differential evolution (CDE), harmony search (HS), genetic algorithm (GA), firefly swarm optimisation (FSO), gravitational search algorithm (GSA), brainstorm optimisation algorithm (BSOA), artificial bee colony (ABC), fixed inertia weight PSO (FIWPSO), chaotic inertia weight PSO (CIWPSO), random inertia weight PSO (RIWPSO), time varying acceleration coefficient PSO (TVACPSO) and constricted PSO (COPSO). The comparisons approve outperformance of ELPSO in all terms of accuracy, scalability and convergence rate.

Appendix A.1.

Pseudocode of ELPSO

Set ELPSO parameters ($C_1, C_2, W_i, W_f, N_p, t_{\max}, V_{\max}, F, h, g$)

Set problem parameters ($n, X_{\min}, X_{\max}, f$) % f stands for objective function which is intended to be minimised.

Initialise swarm randomly, set the objective value of each particle as its personal best and set the best objective value among all particles as swarm best.

$t=0$% t is iteration number
While $t \Leftarrow t_{\max}$ do
For $i=1{:}N_p$ do
% Flight equations
Update the velocities of particles by two following flight equations

$$V_i(t+1)=\omega V_i(t)+C_1r_1(P_i-X_i)+C_2r_2(P_g-X_i)$$

$$X_i(t+1)=X_i(t)+V_i(t+1)$$

Update personal bests and swarm best
% End of flight equations
% Start of five successive mutations on P_g
% Stage 1: Gaussian mutation
For $d=1{:}n$ do
$P_{g1}(d)=P_g(d)+(X_{\max}(d)-X_{\min}(d))\cdot \text{Gaussian}(o,h)$
End for
Bound position of P_{g1}
If $f(P_{g1})<g_{\text{best}}$ do
$P_g=P_{g1}$
$g_{\text{best}}=f(P_{g1})$
End if
$h=h-(1/t_{\max})$
% End of stage 1(Gaussian mutation)
% Stage 2: Cauchy mutation
For $d=1{:}n$ do
$P_{g2}(d)=P_g(d)+(X_{\max}(d)-X_{\min}(d))\cdot \text{Cauchy}(o,s)$
End for
Bound position of P_{g2}
If f (P_{g2}) < g_{best} do
$P_g=P_{g2}$
$g_{\text{best}}=f(P_{g2})$
End if
$s=s-(1/t_{\max})$
% End of stage 2 (Cauchy mutation)
% Stage 3: opposition-based mutation operator separately for each dimension of P_g
For $d=1$:n
$P_{g3}=P_g$
$P_{g3}(d)=X_{\min}(d)+X_{\max}(d)-P_{g3}(d)$
If $f(P_{g3})<f(P_{g3})$
$P_g=P_{g3}$
End if
End for
End
Bound $P_{g3}(d)$ to feasible region
If $f(P_{g3})<g_{\text{best}}$ do
$P_g=P_{g3}$
$g_{\text{best}}=f(P_{g3})$
End if
% End of stage 3 (opposition-based mutation separately for each dimension of P_g)
% Stage 4: opposition-based mutation on P_g as a whole
$P_{g4}=X_{\min}+X_{\min}-P_g$
Bound position of P_{g4}
If $f(P_{g4})<g_{\text{best}}$ do
$P_g=P_{g4}$
$g_{\text{best}}=f(P_{g4})$
End if
% End of stage 4 (opposition-based mutation on P_g as a whole)
% Stage 5 DE-based mutation
$P_{g5}=P_g+F(X_r-X_s)$ % r and s are two random unequal particles in swarm
Bound P_{g5} to feasible region
If $f(P_{g4})<g_{\text{best}}$ do
$P_g=P_{g4}$
$g_{\text{best}}=f(P_{g4})$
End if
% End of stage 5 (DE-based mutation)
% End of comprehensive premature convergence alleviator
$\omega=\omega-(1/t_{\max})$
End for
End while
Print P_g and g_{best}

Appendix A.2.

Table A.1.

Table A.1
Specifications of test functions.

Function name	Formulation	Range
Sphere	$F_1(X)=\sum_{i=1}^{n} X_i^2$	[−5.12, 5.12]
Dixon & price	$F_2(X)=(X_1-1)^2+\sum_{i=2}^{n} i(2X_i^2-X_{i-1})^2$	[−10, 10]

Table A.1 (*Continued*)

Function name	Formulation	Range
Zakharov	$F_3(X) = \sum_{i=1}^{n} X_i^2 + \left(\sum_{i=1}^{n} 0.5iX_i\right)^2 + \left(\sum_{i=1}^{n} 0.5iX_i\right)^4$	[−5, 10]
Rastrigin	$F_4(X) = 10n + \sum_{i=1}^{n} (X_i^2 - 10\cos(2\pi X_i))$	[−5.12, 5.12]
Levy	$F_5(X) = \sin^2(\pi X_i) + \sum_{i=1}^{n-1} (y_i-1)^2[1+10\sin^2(\pi y_i+1)] + (y_i-1)^2[1+\sin^2(2\pi y_i)]$ Where $y_i = 1 + \frac{X_i - 1}{4}$ *for* $i = 1, 2, \ldots, n$	[−15, 30]
Griewank	$F_6(X) = \sum_{i=1}^{n} \frac{X_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	[−600, 600]
Rosenbrock	$F_7(X) = \sum_{i=1}^{n-1} \left[100(X_i^2 - X_{i+1})^2 + (X_i - 1)^2\right]$	[−5, 10]
Ackley	$F_8(X) = 20 + e - 20\exp\left(-0.2\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} X_i^2\right)}\right) - \exp\left(\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi X_i)\right)}\right)$	[−15, 30]

Table A.2
Comparison of ELPSO with other PSO variants for unimodal functions in dimension 5.

	ELPSO	LDIWPSO	FIWPSO	C IWPSO	RIWPSO	TVACPSO	COPSO	CDE	ODE
F1									
Mean	**3.20e−15**	2.176e−13	6.49e−15	2.98e−13	9.07e−13	1.313e−15	1.458e−12	5.13e−15	4.89e−15
Std	**3.10e−15**	1.936e−13	7.20e−15	3.94e−13	2.209e−12	1.024e−15	1.658e−12	4.13e−15	3.65e−15
Median	**2.02e−15**	1.55e−13	3.19e−15	8.6e−14	1.0e−14	1.140e−15	7.70e−13	3.11e−15	2.93e−15
Min	**1.5e−16**	3.43e−14	4.1e−16	1.1e−14	0	1.11e−16	1.56e−13	1.89e−16	4.59e−16
Max	**1.145e−14**	9.984e−13	2.963e−14	1.316e−12	9.967e−12	3.528e−15	6.864e−12	5.64e−14	8.12e−14
F2									
Mean	**5.17e−13**	1.683e−11	7.81e−13	5.89e−10	1.80e−10	1.401e−13	2.367e−10	7.56e−13	9.23e−13
Std	**5.88e−13**	1.892e−11	1.059e−12	1.724e−9	4.63e−10	9.68e−14	2.909e−10	4.33e−13	8.65e−13
Median	**3.08e−13**	1.249e−11	4.47e−13	3.5e−11	7e−12	1.419e−13	9.60e−11	3.66e−13	6.32e−13
Min	**5.3e−14**	1.7e−13	5.9e−14	0	0	3.6e−15	1.19e−11	8.49e−14	9.75e−14
Max	**2.669e−12**	9.100e−11	5.127e−12	7.954e−9	2.137e−9	3.196e−13	9.826e−10	8.21e−12	7.11e−12
F3									
Mean	1.644e−14	9.47e−13	3.456e−14	2.19e−12	5.13e−11	1.228e−14	1.464e−11	**2.35e−15**	2.09e−12
Std	1.494e−14	7.34e−13	2.493e−14	3.82e−12	1.431e−10	1.087e−14	1.790e−11	**8.06e−16**	3.02e−12
Median	1.281e−14	6.15e−13	3.044e−14	9.1e−13	1e−13	8.55e−15	5.99e−12	**9.21e−15**	8.17e−13
Min	1.51e−15	1.69e−13	2.29e−15	2e−14	0	8.9e−16	1.70e−12	**5.41e−16**	1.41e−14
Max	7.225e−14	2.703e−12	9.742e−14	1.517e−11	6.529e−10	4.513e−14	7.895e−11	**2.88e−14**	1.23e−11

Table A.3
Comparison of ELPSO with other PSO variants for multimodal functions in dimension 5.

	ELPSO	LDIWPSO	FIWPSO	C IWPSO	RIWPSO	TVACPSO	COPSO	CDE	ODE
F4									
Mean	**0.4261**	1.2939	2.0573	1.1277	1.9971	1.3930	3.6736	0.8543	0.7568
Std	**0.1931**	0.9111	1.0752	0.9323	1.0643	0.8166	1.7799	0.2658	0.3217
Median	**0.3924**	0.9950	1.9899	0.9950	1.9899	0.9950	2.9894	0.5537	0.6784
Min	**0**	0	0	0	0	0	0.9950	0	0
Max	**1.7849**	3.9812	3.9798	2.9849	4.9748	2.9849	6.9647	3.9798	2.5348
F5									
Mean	1.109e−13	4.44e−13	2.10e−14	7.55e−13	1.56e−11	2.516e−15	4.38e−12	**1.53e−15**	8.23e−14
Std	1.103e−13	4.62e−13	3.22e−14	1.244e−12	7.76e−11	1.950e−15	8.88e−12	**4.54e−16**	9.45e−14
Median	6.70e−14	2.92e−13	9.6e−15	2.20e−13	2e−13	2.128e−15	1.93e−12	**1.21e−15**	1.54e−14
Min	1.16e−14	2.5e−14	3.5e−15	3e−15	0	3.68e−16	3.6e−13	**2.43e−16**	5.57e−15
Max	4.165e−13	2.349e−12	1.586e−13	4.831e−12	4.261e−10	5.969e−15	4.942e−11	**1.74e−14**	7.931e−13
F6									
Mean	**0.0114**	0.0891	0.0783	0.1113	0.0877	0.0757	0.1243	0.0326	0.0648
Std	**0.0039**	0.0411	0.0529	0.0820	0.0350	0.0562	0.0599	0.0163	0.0085
Median	**0.0121**	0.0788	0.0692	0.0899	0.0838	0.0513	0.1059	0.0327	0.0536
Min	**0.0010**	0.0271	0.0101	0.0222	0.0296	0.0106	0.0172	0.0163	0.0103
Max	**0.0246**	0.1577	0.2290	0.3869	0.1700	0.2020	0.2660	0.0754	0.1162
F7									
Mean	**0.0952**	0.1352	0.5241	0.2744	0.6876	0.7862	0.6551	0.1432	0.2663
Std	**0.1988**	0.7172	1.3591	0.9962	1.4858	1.6132	1.4900	0.0843	0.1965
Median	**0.0821**	0.3321	0.3426	0.2576	0.5584	0.6548	0.4584	0.1348	0.2547
Min	**0**	0	0	0	0	0	0	0	0
Max	**0.9308**	3.9308	3.9308	3.9308	3.9308	3.9308	3.9308	0.9308	3.9308

Table A.3 (*Continued*)

	ELPSO	LDIWPSO	FIWPSO	C IWPSO	RIWPSO	TVACPSO	COPSO	CDE	ODE
F8									
Mean	**1.62e−7**	3.012e−6	6.42e−7	3.44e−6	4.31e−6	2.731e−7	1.066e−5	2.25e−7	7.21e−6
Std	**2.57e−8**	1.608e−6	3.09e−7	2.91e−6	7.15e−6	8.04e−8	8.13e−6	7.14e−8	4.15e−6
Median	**1.30e−7**	2.349e−6	6.15e−7	2.68e−6	1.49e−6	2.681e−7	9.55e−6	2.08e−7	4.39e−6
Min	**1.65e−8**	9.35e−7	2.43e−7	2.7e−7	2e−8	1.288e−7	3.11e−6	4.23e−8	2.06e−8
Max	**8.137e−7**	6.509e−6	1.598e−6	1.160e−5	2.921e−5	4.239e−7	3.596e−5	9.94e−7	1.621e−5

Table A.4
Comparison of ELPSO with other PSO variants for unimodal functions in dimension 30.

	ELPSO	LDIWPSO	FIWPSO	C IWPSO	RIWPSO	TVACPSO	COPSO	CDE	ODE
F1									
Mean	**5.244e−8**	6.11e−8	1.701e−7	1.565e−7	4.81e−7	1.077e−9	9.55e−9	8.13e−8	7.54e−8
Std	**1.643e−8**	3.64e−8	1.083e−7	7.64e−8	7.24e−7	4.14e−10	3.47e−9	2.16e−8	2.85e−8
Median	**5.176e−8**	5.92e−8	1.173e−7	1.676e−7	6.9e−8	1.046e−9	9.47e−9	6.94e−8	6.11e−8
Min	**2.979e−8**	1.36e−8	7.28e−8	5.40e−8	1.6e−8	4.43e−10	5.08e−9	3.32e−8	4.74e−8
Max	**7.614e−8**	1.500e−7	4.254e−7	2.989e−7	1.921e−6	1.921e−9	1.754e−8	9.22e−8	9.93e−8
F2									
Mean	**0.6174**	0.6919	0.6855	0.6731	0.6998	0.6758	0.6670	0.7158	0.6391
Std	**0.1204**	0.0691	0.0437	0.0103	0.0673	0.0241	0.0008	0.0241	0.0461
Median	**0.5667**	0.6689	0.6690	0.6693	0.6701	0.6670	0.6667	0.6910	0.6690
Min	**0.2169**	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667	0.6667
Max	**0.9504**	0.9988	0.8078	0.7010	0.8665	0.7441	0.6700	0.7541	0.7838
F3									
Mean	0.0003	0.0005	0.0032	0.0030	0.0019	1.356e−4	6.9459	4.37e−5	**2.34e−5**
Std	0.0003	0.0003	0.0021	0.0027	0.0020	6.69e−5	10.4052	7.11e−6	**3.11e−6**
Median	0.0002	0.0004	0.0028	0.0021	0.0012	1.365e−4	2.1362	5.21e−5	**3.47e−5**
Min	0.0001	0.0001	0.0008	0.0010	0.0003	4.95e−5	0.0524	1.18e−5	**1.94e−6**
Max	0.001	0.0015	0.0074	0.0098	0.0073	2.708e−4	32.7698	5.39e−4	**2.44e−4**

Table A.5
Comparison of ELPSO with other PSO variants for multimodal functions in dimension 30.

	ELPSO	LDIWPSO	FIWPSO	C IWPSO	RIWPSO	TVACPSO	COPSO	CDE	ODE
F4									
Mean	**8.6403**	25.6700	29.5503	19.0042	25.5707	24.7745	51.6870	9.2553	11.4385
Std	**4.1871**	5.3637	4.9648	7.7839	9.0228	7.6417	16.3067	4.9452	5.9733
Median	**8.8185**	26.8639	28.3564	19.8992	25.3715	25.3714	47.7580	6.1234	6.1173
Min	**3.8941**	13.9294	22.8841	7.9597	11.9397	9.9496	30.8437	4.1146	4.5638
Max	**18.8062**	33.8287	40.7934	32.8337	37.8106	36.8134	92.5310	20.3450	21.6479
F5									
Mean	0.1986	0.4689	0.2783	0.1351	0.2072	0.4875	1.4620	0.1432	**0.1047**
Std	0.0717	0.4229	0.2147	0.1352	0.1532	0.3374	1.5974	0.0846	**0.0648**
Median	0.1087	0.3581	0.2686	0.0898	0.1791	0.4062	0.8124	0.0945	**0.0854**
Min	0	0	0.0001	0	0.0059	0.0895	0.0895	0	**0**
Max	0.5420	1.6360	0.8124	0.4484	0.5439	1.0877	6.5397	0.5369	**0.4173**
F6									
Mean	**2.748e−4**	0.0135	0.0118	0.0124	0.0114	0.0209	0.0136	3.22e−4	4.69e−4
Std	**1.232e−4**	0.0130	0.0089	0.0065	0.0115	0.0231	0.0125	1.93e−4	1.31e−4
Median	**2.547e−4**	0.0089	0.0097	0.0119	0.0083	0.0148	0.0136	4.84e−4	3.83e−4
Min	**1.628e−4**	0.0002	0.0007	0.0015	0	0	0	0	3.45e−4
Max	**4.068e−4**	0.0520	0.0273	0.0247	0.0317	0.0638	0.0369	7.12e−4	8.21e−4
F7									
Mean	5.8172	52.7099	52.5216	8.6884	28.3862	39.3228	45.4664	5.8520	**5.3219**
Std	1.3840	31.4375	26.4630	23.2933	24.2830	30.5514	30.1859	1.7453	**1.2138**
Median	5.8938	29.2173	49.8166	0.5610	24.8883	27.2525	45.7893	6.1231	**4.9873**
Min	1.1337	4.4676	26.9238	0.0186	0.2243	0.0024	1.2764	1.9458	**1.0968**
Max	9.4242	110.5213	86.0921	74.7777	74.0913	85.1819	84.1444	10.4366	**8.2144**
F8									
Mean	**0.2779**	1.8896	2.6922	2.5694	2.3145	2.1255	2.9663	0.4657	0.5376
Std	**0.0816**	0.5905	0.7910	0.9076	0.8196	0.5252	0.7933	0.2154	0.1057
Median	**0.3169**	2.0099	2.4962	2.4482	2.3584	2.2210	2.8098	0.3450	0.2854
Min	**0.0780**	0.0009	1.1552	1.3567	0.9313	1.1570	2.1201	0.1169	0.1349
Max	**0.7389**	2.8141	4.0761	4.1227	3.5214	3.0937	4.9546	0.9463	0.9150

Table A.6
Wilcoxon test results for comparing ELPSO with modified variants of DE and PSO considering all dimensions.

ELPSO vs	LDIWPSO	FIWPSO	C IWPSO	RIWPSO	TVACPSO	COPSO	CDE	ODE
	4.3778e−4	7.7641e−4	0.0027	4.3778e−4	0.0174	0.0011	0.0262	0.0980

Appendix A.3.

Tables A.2–A.6.

References

[1] A. Rezaee Jordehi, J. Jasni, N.I. Abdul Wahab, A. Kadir, M.Z. Abidin, Particle swarm optimisation applications in FACTS optimisation problem, in: Power Engineering and Optimization Conference (PEOCO), 2013 IEEE 7th International, IEEE, 2013, pp. 193–198.

[2] A. Rezaee Jordehi, J. Jasni, A comprehensive review on methods for solving FACTS optimization problem in Power Systems, Int. Rev. Electr. Eng. 6 (2011).

[3] Z. Beheshti, S.M. Hj Shamsuddin, CAPSO: a centripetal accelerated particle swarm optimization, Inf. Sci. 258 (2014) 54–79.

[4] A. Rezaee Jordehi, J. Jasni, Parameter selection in particle swarm optimisation: a survey, J. Exp. Theor. Artif. Intell. 25 (2013) 527–542.

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

[6] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, IEEE, 1998, pp. 69–73.

[7] A. Rezaee Jordehi, J. Jasni, Particle swarm optimisation for discrete optimisation problems: a review, Artif. Intell. Rev. (2013) 1–16.

[8] A. Rezaee Jordehi, J. Jasni, N. Abd Wahab, M.Z. Kadir, M.S. Javadi, Enhanced leader PSO (ELPSO): a new algorithm for allocating distributed TCSC's in power systems, Int. J. Electr. Power Energy Syst. 64 (2015) 771–784.

[9] A. Rezaee Jordehi, Particle swarm optimisation for dynamic optimisation problems: a review, Neural Comput. Appl. (2014) 1–10.

[10] A. Rezaee Jordehi, J. Jasni, Approaches for FACTS optimization problem in power systems, in: IEEE International Power Engineering and Optimization Conference (PEOCO) Melaka, Malaysia, 2012, pp. 355–360.

[11] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, in: IEEE Transactions on Evolutionary Computation, vol. 6, 2002, pp. 58–73.

[12] M.S. Kıran, M. Gündüz, A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems, Appl. Soft Comput. 13 (2013) 2188–2203.

[13] F. Neri, E. Mininno, G. Iacca, Compact particle swarm optimization, Inf. Sci. 239 (2013) 96–121.

[14] H. Wang, H. Sun, C. Li, S. Rahnamayan, J.-s. Pan, Diversity enhanced particle swarm optimization with neighborhood search, Inf. Sci. 223 (2013) 119–135.

[15] Y. Shi, H. Liu, L. Gao, G. Zhang, Cellular particle swarm optimization, Inf. Sci. 181 (2011) 4460–4493.

[16] Y. Wang, B. Li, T. Weise, J. Wang, B. Yuan, Q. Tian, Self-adaptive learning based particle swarm optimization, Inf. Sci. 181 (2011) 4515–4538.

[17] H. Huang, H. Qin, Z. Hao, A. Lim, Example-based learning particle swarm optimization for continuous optimization, Inf. Sci. 182 (2012) 125–138.

[18] M.M. Noel, A new gradient based particle swarm optimization algorithm for accurate computation of global minimum, Appl. Soft Comput. 12 (2012) 353–359.

[19] M.-R. Chen, X. Li, X. Zhang, Y.-Z. Lu, A novel particle swarm optimizer hybridized with extremal optimization, Appl. Soft Comput. 10 (2010) 367–373.

[20] I. Ziari, A. Jalilian, Optimal placement and sizing of multiple APLCs using a modified discrete PSO, Int. J. Electr. Power Energy Syst. 43 (2012) 630–639.

[21] A. Mahor, S. Rangnekar, Short term generation scheduling of cascaded hydro electric system using novel self adaptive inertia weight PSO, Int. J. Electr. Power Energy Syst. 34 (2012) 1–9.

[22] V. Hosseinnezhad, E. Babaei, Economic load dispatch using θ-PSO, Int. J. Electr. Power Energy Syst. 49 (2013) 160–169.

[23] M. Eslami, H. Shareef, A. Mohamed, M. Khajehzadeh, An efficient particle swarm optimization technique with chaotic sequence for optimal tuning and placement of PSS in power systems, Int. J. Electr. Power Energy Syst. 43 (2012) 1467–1478.

[24] S. Panda, N.K. Yegireddy, S.K. Mohapatra, Hybrid BFOA–PSO approach for coordinated design of PSS and SSSC-based controller considering time delays, Int J. Electr. Power Energy Syst. 49 (2013) 221–233.

[25] M.-T. Vakil Baghmisheh, M. Peimani, M.H. Sadeghi, M.M. Ettefagh, A.F. Tabrizi, A hybrid particle swarm–Nelder–Mead optimization method for crack detection in cantilever beams, Appl. Soft Comput. 12 (2012) 2217–2226.

[26] M. Tang, Y. Xin, J. Li, J. Zhai, Nonconvex resource control and lifetime optimization in wireless video sensor networks based on chaotic particle swarm optimization, Appl. Soft Comput. 13 (2013) 3273–3284.

[27] M. Joorabian, E. Afzalan, Optimal power flow under both normal and contingent operation conditions using the hybrid fuzzy particle swarm optimisation and Nelder–Mead algorithm (HFPSO–NM), Appl. Soft Comput. 14 (2014) 623–633.

[28] L. Idoumghar, N. Chérin, P. Siarry, R. Roche, A. Miraoui, Hybrid ICA–PSO algorithm for continuous optimization, Appl. Math. Comput. 219 (2013) 11149–11170.

[29] A. Gálvez, A. Iglesias, A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing, Appl. Soft Comput. 13 (2013) 1491–1504.

[30] J. Behnamian, S. Fatemi Ghomi, Development of a PSO–SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting, Expert Syst. Appl. 37 (2010) 974–984.

[31] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.

[32] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (1988) 95–99.

[33] X.-S. Yang, Firefly algorithm, Levy flights and global optimization, in: Research and Development in Intelligent Systems XXVI, Springer, London, 2010, pp. 209–218.

[34] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.

[35] Y. Shi, An optimization algorithm based on brainstorming process, Int. J. Swarm Intell. Res. (IJSIR) 2 (2011) 35–62.

[36] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (2007) 459–471.

[37] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, Opposition-based differential evolution, in: IEEE Transactions on Evolutionary Computation, vol. 12, 2008, pp. 64–79.

[38] S. Rahnamayan, H.R. Tizhoosh, M. Salama, Opposition versus randomness in soft computing techniques, Appl. Soft Comput. 8 (2008) 906–918.

[39] J. Tvrdík, R. Polakova, Competitive differential evolution applied to CEC 2013 problems, in: IEEE Congress on Evolutionary Computation (CEC), 2013, IEEE, 2013, pp. 1651–1657.

[40] A.R. Jordehi, M. Joorabian, Optimal placement of Multi-type FACTS devices in power systems using evolution strategies, in: Power Engineering and Optimization Conference (PEOCO), 2011 5th International, IEEE, 2011, pp. 352–357.

[41] A. Rezaee Jordehi, Chaotic Bat Swarm Optimisation (CBSO), Appl. Soft Comput. (2014).

[42] A. Rezaee Jordehi, A chaotic-based big bang-big crunch algorithm for solving global optimisation problems, Neural Comput. Appl. (2014).

[43] A. Rezaee Jordehi, A chaotic artificial immune system optimisation algorithm for solving global continuous optimisation problems, Neural Comput. Appl. (2014).

[44] A. Rezaee Jordehi, Heuristic methods for solution of FACTS optimization problem in power systems, in: 2011 IEEE Student Conference on Research and Development, 2011, pp. 30–35.