Particle swarm optimisation with time varying cognitive avoidance component

Anupam Biswas* and Bhaskar Biswas

Department of Computer Science and Engineering,
Indian Institute of Technology (BHU),
Varanasi, U.P., India
Email: abanumail@gmail.com
Email: bhaskar.cse@iitbhu.ac.in
*Corresponding author

Anoj Kumar and K.K. Mishra

Department of Computer Science and Engineering,
Motilal Nehru National Institute of Technology,
Allahabad, U.P., India
Email: anojk@mnnit.ac.in
Email: kkm@mnnit.ac.in

Abstract: Interactive cooperation of local best or global best solutions encourages particles to move towards them, hoping that better solution may present in the neighbouring positions around local best or global best. This encouragement does not guarantee that movements taken by the particles will always be suitable. Sometimes, it may mislead particles in the wrong direction towards the worst solution. Prior knowledge of worst solutions may predict such misguidance and avoid such moves. The worst solution cannot be known in prior and can be known only by experiencing it. This paper introduces a cognitive avoidance scheme to the particle swarm optimisation method. A very similar kind of mechanism is used to incorporate worst solutions into strategic movement of particles as utilised during incorporation of best solutions. Time varying approach is also extrapolated to the cognitive avoidance scheme to deal with negative effects. The proposed approach is tested with 25 benchmark functions of CEC 2005 special session on real parameter optimisation as well as with four other very popular benchmark functions.

Keywords: optimisation; particle swarm optimisation; PSO; differential evolution; heuristics.

Reference to this paper should be made as follows: Biswas, A., Biswas, B., Kumar, A. and Mishra, K.K. (2018) 'Particle swarm optimisation with time varying cognitive avoidance component', *Int. J. Computational Science and Engineering*, Vol. 16, No. 1, pp.27–41.

Biographical notes: Anupam Biswas is pursuing his PhD in Computer Science and Engineering from Indian Institute of Technology (BHU), Varanasi. He received his MTech in Computer Science and Engineering from Motilal Nehru National Institute of Technology Allahabad. His research interests include evolutionary computation, optimisation, data mining, and social network analysis.

Bhaskar Biswas received his PhD in Computer Engineering from Indian Institute of Technology (BHU), Varanasi. He is working as an Assistant Professor in the same Department. His research interests include data mining, web mining, wireless sensor network and social network analysis, and evolutionary computation.

Anoj Kumar is pursuing his PhD in Computer Science and Engineering from Motilal Nehru National Institute of Technology Allahabad. His research interests are related to optimisation, software testing, and computer graphics.

K.K. Mishra received his PhD in Computer Science and Engineering from Motilal Nehru National Institute of Technology Allahabad. He is working as and Assistant Professor in the same department. His research area covers evolutionary computing, software testing, analysis of algorithm, automata theory, microprocessor, and multi-objective optimisation.

This paper is a revised and expanded version of a paper entitled 'Particle swarm optimisation with cognitive avoidance component' presented at the 2nd International Conference on Advances in Computing, Communication and Informatics (ICACCI 2013), Mysore, India, 22–25 August 2013.

1 Introduction

Application domain of swarm intelligent techniques has grown extensively in last two decades. Several domains including image processing (Liu et al., 2011), sociological analysis (Honghao et al., 2013), power system (Rajagopalan and Mala, 2014; Sharma et al., 2014), fuzzy system (Khosla et al., 2014), signal processing (Zhang et al., 2013), process scheduling (Chen et al., 2014), etc. have adopted such techniques mainly for optimisation purpose. With diverse application domains most of these techniques have been modified to get better result. Particle swarm optimisation (PSO) is one such technique. Despite numerous modifications, some of the problems still remain with the approach adopted.

PSO is a very successful algorithm of last two decades for solving optimisation problems, and originally proposed in Kennedy and Eberhart (1995), Eberhart and Kennedy (1995), and Eberhart et al. (1996). Main inspiration behind the PSO algorithm is social behaviour of swarms such as birds flocking, fish schooling. Similar to other population-based algorithms such as genetic algorithm (GA) (Holland, 1975; Goldberg, 1989, 1990; Srinivas and Patnaik, 1994), PSO also maintains a population of particles referred as the swarm. Individual solution is considered as particles in the swarm. Each particle interacts with the others and simultaneously learns from its own experience. Similar to GA, the suitability of a particle is defined by its ability to survive in the solution domain, i.e., fitness. As PSO learns from its neighbour so, particles with higher fitness put comparatively greater impact on swarm behaviour. Each particle is associated with a velocity and a position in the solution domain. Every particle maintains its best solution experienced so far, i.e., the local best (pbest) and best solution experienced by neighbour, i.e., the global best (gbest). Positions and velocities are updated in accordance with current pbest and gbest.

Position, velocity, pbest and gbest vector of i^{th} particle at t^{th} iteration in d dimension can be represented respectively as shown below:

$$X_i(t) = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id})$$
$$V_i(t) = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{id})$$
$$P_i(t) = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{id})$$
$$G_i(t) = (g_{i1}, g_{i2}, g_{i3}, \ldots, g_{id})$$

Velocity and position of i^{th} particle in j^{th} dimension for next iteration are evaluated and updated with the following two equations:

$$\begin{aligned} V_{ij}(t+1) = V_{ij}(t) + C_1 \times R_1 \times (P_{ij}(t) - X_{ij}(t)) \\ +C_2 \times R_2 \left(G_{ij}(t) - X_{ij}(t)\right) \end{aligned} \tag{1}$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \tag{2}$$

Here, R_1 and R_2 are uniformly distributed random numbers in range [0, 1]. C_1 and C_2 are the positive constants in range (0, 2], known as acceleration coefficients. C_1 controls particle's movement towards local best and C_2 controls particle's movement towards global best. The term $C_1 \times R_1 \times (P_{ij}(t) - X_{ij}(t))$ is associated with particle's cognition of its own best solution. The term $C_2 \times R_2 \times (G_{ij}(t) - X_{ij}(t))$ is associated with particle's collaborative interaction with its neighbours. These two terms are related with particle's acceleration (rate of change in velocity) so these are often known as cognitive acceleration and social acceleration respectively. In this paper, another component, called cognitive avoidance, is introduced.

The rest of this paper is organised as follows: Section 2 briefed variants of PSO and motive behind such variations, also our contribution to this paper. Section 3 describes the main motivation of our proposal. Section 4 illustrates our proposed approach in detail. Section 5 discusses performance of proposed approaches with benchmark functions. Finally, concluded in Section 6.

2 PSO variants

2.1 Binary PSO

In binary PSO (Tasgetiren and Lian, 2004; Gong and Tuson, 2007; Chuang et al., 2008), each particle represents its position in binary values which are 0 or 1. Decisions taken by each particles in the population is binary, i.e., either YES/TRUE = 1 or NO/FALSE = 0. The velocity vector equation and position vector equation are defined as:

$$V_i(t+1) = \frac{1}{1+e^{-V_i(t)}} \tag{3}$$

$$X_i(t+1) = \begin{cases} 1 & \text{if } r < V_i(t+1) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where r is a uniform random number in the range [0, 1].

2.2 Discrete PSO

In discrete PSO, particle's positions (Parsopoulos and Vrahatis, 2002) are discrete values. The algorithm uses real (discrete) values velocities and positions. Discrete PSO has a high success rate in solving integer programming problems and has the ability to quickly converge towards the optimal solution (Shayeghi et al., 2010).

2.3 Constricted PSO

Velocity of particle is cannot be controlled in the standard version of PSO. Hence, there is a possibility that sometime particles may achieve very high velocity and move outside the search space. Kennedy et al. (2001) introduce a mechanism to control velocities of particles by limiting them to $V_{\max}$. Clerc and Kennedy (2002) introduce constriction parameter χ equivalent to the $V_{\max}$ parameter. Several types of constrictions such as Type 1, Type 1′ and Type 1″ have been proposed in Clerc and Kennedy (2002). Type 1″ constriction is simplest among all other constrictions. In this type of PSO, velocity is updated as follows:

$$V_i(t+1) = \chi\big(V_i(t) + U[0, \varphi_1](P_i(t) - X_i(t)) \\ + U[0, \varphi_2](G_i(t) - X_i(t))\big) \tag{5}$$

where $\chi = \frac{2k}{2-\varphi-\sqrt{\varphi^2-4\varphi}}$, here $k \in [0, 1]$, $\varphi = \varphi_1 + \varphi_2$, $\varphi > 4$. Generally, the value of φ is considered so as $\varphi_1 = \varphi_2 = 2.05$ and $k = 1$ which results $\chi \approx 0.729$.

2.4 Fully informed particle swarm

In classical PSO as well as in constricted PSO, velocity of a particle is influenced only by particle's personal best and global best. It seems that other knowledge available to neighbour has remain unused. As far as social behaviour is concern, generally shares information of all individuals. Mendes (2004) has proposed fully informed particle swarm (FIPS) considering all information of neighbour to update velocity. In FIPS, velocity is updated as follows:

$$V_i(t+1) = \chi\left(V_i(t) + \frac{\sum_{k\in N} U[0, \varphi_{\max}](P_k(t) - X_i(t))}{N}\right) \tag{6}$$

where N is the swarm size, $P_k(t)$ is the personal best value of neighbour k at time t.

2.5 Orthogonal learning PSO

In traditional PSO learns from best values experienced summing linearly. Though it is very easy to build-up such strategy, it is not so efficient for the complex system. Zhan et al (2011), and Zhan and Zhang (2011) proposed a mechanism which utilises orthogonal learning strategy in the traditional PSO. The strategy incorporates orthogonal experimental design (OED) (MSR Group, 1975; Montgomery, 2000). OED is used to discover the best combination of a particle's personal best position and global best position. Best combinations of sample position are generated with OED method, which allows particles to fly more steadily.

2.6 Parameter tuning

In original PSO main challenge was to choose appropriate parameters for efficient performance and this has been studied since introduction of PSO. The primary focus of such study is inertia weight, cognitive acceleration coefficient and social acceleration coefficient. These parameters are tuned for improving the optimal solution in the solution domain. Parameter tuning strategy is needed because the basic version of the PSO was very effective on some problems. Proper and fine tuning of the parameters shows improvement in result (Arya et al., 2014).

Exploration of solution space means to search solution space far from the current position covering entire solution space, while exploitation means to look around the current position and to cover smaller area surrounding present location (Črepinšek et al., 2013). Initially, the values of the parameters of PSO algorithm were constant, which results in imbalanced exploration and exploitation. However, experimental results proved that it is better to initially set the parameters to a large value, in order to promote global exploration of the search space, and gradually decrease to get more refined optimal solutions. A large parameter value facilitates global exploration, while a small one tends to facilitate exploitation. A suitable value for the parameters usually provides balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution. Most prominent parameter tuning approaches are briefed below.

2.6.1 PSO – time varying inertia weight (PSO-TVIW)

A large inertia weight results a global search while a small inertia weight results a local search. Shi and Eberhart (1999) have linearly decreases weight from larger value implies global search ability at the beginning and local search ability at the end. The inertia weight changes with the following equation:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min})\frac{I_{\max} - I_{current}}{I_{\max}} \tag{7}$$

where $\omega_{\min}$ and $\omega_{\max}$ are the minimum value and maximum value of weight respectively. $I_{\max}$ is the maximum number of iterations and $I_{current}$ is the present iteration number. Shi and Eberhart (1999) show $\omega_{\min} = 0.4$ and $\omega_{\max} = 0.9$ shown better performance.

2.6.2 PSO-random inertia weight (PSO-RANDIW)

It cannot be predicted whether exploration is for a larger inertia weight value or exploitation is for a smaller inertia weight. It may be better at any given time for exploration or exploitation irrespective of smaller or larger weight. Eberhart and Shi (2001) proposed random inertia weight instead of gradually decreasing or any static value. The inertia weight changes randomly according to the following equation:

$$\omega = 0.5 + \frac{r}{2} \tag{8}$$

where r is any random value in between 0.5 to 1.

2.6.3 PSO – time varying acceleration coefficients (PSO-TVAC)

With extension to PSO-TVIW, Ratnaweera et al. (2004) introduce linearly varying acceleration coefficients C_1 and C_2 along with ω. This additional linearity to PSO-TVIW became more effective to global search during beginning of the algorithm and to local search at the end. Parameters of PSO-TVAC are updated as follows:

$$C_1 = C_{1\min} + \left(C_{1\max} - C_{1\min}\right)\left(\frac{I_{\max} - I_{current}}{I_{\max}}\right) \tag{9}$$

$$C_2 = C_{2\max} + \left(C_{2\min} - C_{2\max}\right)\left(\frac{I_{\max} - I_{current}}{I_{\max}}\right) \tag{10}$$

$$\omega = \omega_{\min} + \left(\omega_{\max} - \omega_{\min}\right)\left(\frac{I_{\max} - I_{current}}{I_{\max}}\right) \tag{11}$$

where $C_{1\min}$ is the minimum value of C_1, $C_{1\max}$ is the maximum value of C_1, $C_{2\min}$ is the minimum value of C_2 and $C_{2\max}$ is the maximum value of C_2. Ratnaweera et al. (2004) show the algorithm performs better when C_1 varies from 2.5 to 0.5, C_2 from 0.5 to 2.5 and ω from 0.9 to 0.4.

2.7 Our contributions

All the variants of PSO briefed above, considers either personal best or global best solution for making strategy on next move. During movement of each particle, it experiences both good as well as bad solution. Strategy made on the basis of good solution may mislead particles. Moreover, part of experienced knowledge remains unutilised and gets wasted if only one kind of solution is used. To meet the requirements of knowledge empowerment and utilisation of both kind of knowledge, we proposed an avoidance scheme along with the best solution strategy. The approach incorporates bad solutions experienced by the particle for making strategy on next move. Instead of following only good solution, the approach uses bad solution to discourage particles to not follow such solution.

This paper has been significantly extended from previous work (Biswas et al., 2013), which discussed preliminary work and the technical aspects of the proposed scheme. The work introduces cognitive avoidance scheme to the standard PSO. Here represents latest results and discussed another aspect of such avoidance scheme. This work enhances cognitive avoidance scheme varying with time, discussed in coming section. Result evaluation is extended to 25 benchmark functions that were presented in CEC 2005 special session (Suganthan et al., 2005).

3 Motivation

Awareness of a particle's own best position (pbest) and neighbour's best position (gbest) helps particles to decide which direction to move and is suitable for convergence. Each particle tracks pbest and gbest in successive iterations. Strategic decision regarding movement of any particle is made based on these two known values. These known best positions attract particles towards themselves with a hope that neighbouring position of best solution will as good as or better than the best one. Indeed, sometimes it may happen that these attractions act as traps for particles. This is because solution nearby current pbest or gbest is not always good, as it is actually depends on the solution domain. Though pbest and gbest influence particle's movement, actual movement is affected by two random values R_1 and R_2. However, this interference is necessary for the algorithm to improve overall exploration of the solution domain.

Despite the fact that pbest and gbest motivate particles to move towards optimal solution, interference of random values may cause particles to move to some unfruitful positions. These unfruitful movements may degrade the algorithm's overall performance. Any wrong movement of a particle may lead to divergence from optimal solution, which may delay convergence. Effect of such unfruitful movement propagated in successive iterations resulting in extra iterations to reach the destination or may diverted to completely different direction. Though such effect may reduce subsequent iterations, the effect of unnecessary movement is always there, causing extra iterations to attain same position. Awareness of such pitfalls may improve overall performance of PSO by avoiding them.

Figure 1 Effect of cognitive avoidance in PSO

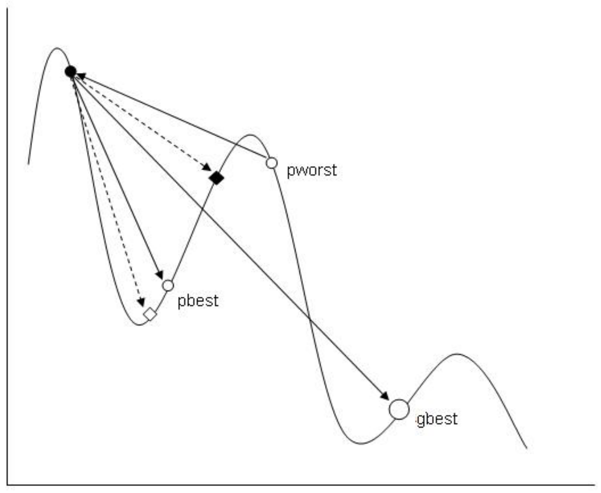


Table 1 Benchmark functions

Func	*Definition*
Rastrigin	$f(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$
Ackley	$f(x)=(20+e)-20\exp\left[-0.2\sqrt{\frac{1}{2}\sum_{i=1}^{n}x_i^2}\right]-\exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right]$
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$
Schwefel	$f(x)=\sum_{i=1}^{n}\left(x_i\sin\sqrt{\lvert x_i\rvert}\right)$

Table 2 Initial range and optima

Func	*Range*	*Optimal solution*
Rastrigin	[–5.12, 5.12]	$f(x^*) = 0$
Ackley	[–15, 30]	$f(x^*) = 0$
Rosenbrock	[–2.048, 2.048]	$f(x^*) = 0$
Schwefel	[–500, 500]	$f(x^*) = -n \times 418.9829$

It is impossible to assure whether the next position is good or bad, it can only be predicted probabilistically depending on previous and present situation. Tracking of pbest and gbest is the best predictive notion incorporated in PSO and which motivates each particle to make movement nearby them considering that the probable good solution might be around the pbest or gbest. With this little greedy approach PSO works very well but as mentioned above every solution around the pbest or gbest is not always good resulting in unbeatable consequences. To avoid such situations, very similar mechanism that pbest and gbest does to a particle to attract towards themselves can be used by making particles aware of such unfruitful movements.

4 Cognitive avoidance scheme

As the nature of solution space is unknown to the particles so the only possibility is to predict next good movements ensuring that particles are moving in the right direction. In PSO convergence towards the optimal solution is guided by two acceleration components (cognitive component and social component). However, it cannot be assured 100% that next position is better. This is only an assumption that solutions near pbest or gbest may be better. There is always a possibility of attaining bad solution due to interference of random parameters as described in previous section. Any misguidance may slow down overall convergence of algorithm towards the optimal solution. Therefore, it is very crucial to handle properly this kind of discrepancy to guide particles in appropriate direction to enhance efficiency and accuracy. Considering this issue, in this paper we have proposed an approach called cognitive avoidance scheme to avoid such situations.

In our proposed approach, each particle maintains its worst value that it has attained so far along with the pbest and gbest. With this known worst value particles try to avoid further movement towards it, having the sense that solutions nearby the worst one may not be suitable. Particles' own known worst solutions so far (pworst) pushes particles backward so that they can never be trapped again into it. This avoidance mechanism also reduces movement of particles towards other bad solutions around the pworst. This is a kind of inverse greedy approach where particles are distracted instead of attracting as pbest and gbest does. To define such avoidance scheme we have added a new component called cognitive avoidance component to the existing velocity equation of PSO. Current pworst vector of i^{th} particle can be represented as $W_i(t) = (w_{i1}, w_{i2}, w_{i3}, \ldots, w_{id})$ where, d is the dimension of particle. Velocity equation is redefined as follows:

$$\begin{aligned} V_i(t+1) = \omega * V_i(t) + C_1 * R_1 * \left(P_i(t) - X_i(t)\right) \\ + C_2 * R_2 * \left(G_i(t) - X_i(t)\right) \\ - C_3 * R_3 \left(W_i(t) - X_i(t)\right) \end{aligned} \tag{12}$$

Table 3 Comparison of PSO and PSOCA

Objective function	*Dimension*	*Measures*	*PSO*	*PSOCA*
Rastrigin's function	10	Mean	6.268238	5.890155
		Std. deviation	3.205177	2.725471
	20	Mean	35.679164	30.087520
		Std. deviation	11.133938	7.594830
	30	Mean	83.605148	80.671492
		Std. deviation	19.746506	6.351290
Ackley's function	10	Mean	0.000000	0.000000
		Std. deviation	0.000000	0.000000
	20	Mean	0.701055	0.511282
		Std. deviation	0.821616	0.790906
	30	Mean	2.379818	2.202456
		Std. deviation	0.878856	0.045834
Rosenbrock's function	10	Mean	0.495389	0.587538
		Std. deviation	1.304553	1.397056
	20	Mean	12.151854	9.016740
		Std. deviation	9.862681	3.645567
	30	Mean	35.199659	33.586877
		Std. deviation	23.205804	23.058436
Schwefel's function	10	Mean	–3,726.731438	–3,733.443933
		Std. deviation	186.940914	175.685358
	20	Mean	–6,312.808570	–6,388.208710
		Std. deviation	393.573309	386.027770
	30	Mean	–8,259.610620	–8,280.060558
		Std. deviation	584.277113	507.231340

Table 4 Comparison of PSO-CATV with PSO-TVIW, PSO-TVAC and jDE part I

F	*Dim*	*Measures*	*PSO-TVIW*	*PSO-TVAC*	*PSO-CATV*	*jDE*
f1	50	Mean	3.902552	4.908112	0.000000	0.000000
		SD	12.940261	8.163804	0.000000	0.000000
	60	Mean	2.196993	8.109815	0.000000	0.000000
		SD	3.415574	16.602715	0.000000	0.000000
	70	Mean	4.089994	7.808844	0.000000	0.000000
		SD	11.795045	16.735104	0.000000	0.000000
f2	50	Mean	130.631028	70.339226	5.745377	11.48375
		SD	41.359146	31.479238	3.555487	4.28824
	60	Mean	329.163198	162.796134	17.549255	183.68
		SD	89.857498	59.878654	10.952076	47.192
	70	Mean	641.125525	413.048155	59.965942	132.897
		SD	153.346504	153.109885	39.948230	54.55045
f3	50	Mean	10,808,432.435	11,897,240.811	3,045,488.056	1,570,000.00
		SD	6,285,139.025	7,327,036.636	1,146,567.177	42,426.40687
	60	Mean	14,087,070.935	11,252,155.437	4,242,028.077	4,355,950
		SD	6,878,314.411	7,192,879.592	1,550,749.991	888,621.091
	70	Mean	16,878,072.985	14,521,790.405	5,933,886.299	5,425,700.00
		SD	10,255,231.182	8,592,850.856	2,364,138.274	1,423,123.108
f4	50	Mean	3,152.656009	7,947.115621	3,566.457645	6107.6
		SD	1,128.932254	2,596.575130	13,858.383438	431.9008219
	60	Mean	6,632.142723	14,127.583262	8,029.763677	18038
		SD	1,842.963164	4,381.406693	3,078.674737	9,916.465499
	70	Mean	12,292.978720	21,543.842494	14,205.333739	25,808
		SD	3,372.047619	4,443.916660	4,162.635158	4,449.1158
f5	50	Mean	7,995.021248	8,232.194651	7,328.712549	5722.7
		SD	1,073.176077	1,444.619689	1,167.819993	771.73634
	60	Mean	10,263.928239	10,644.165517	10,542.227027	12,593.6
		SD	1,878.519310	1,510.495959	1,925.873026	4,499.179
	70	Mean	13,694.404809	13,451.695247	11,685.651189	13,239.5
		SD	2,318.435464	2,431.814152	1,760.678965	178.89801
f6	50	Mean	212.248266	163.344817	96.338635	297.845
		SD	253.610211	126.706071	89.031715	137.89289
	60	Mean	108.252552	122.628378	73.948984	161.995
		SD	62.138997	43.533576	35.277724	4.3345
	70	Mean	135.399165	187.695732	110.006910	173.225
		SD	60.001736	114.768624	56.843038	18.6039
f7	50	Mean	0.995867	0.994219	0.008458	6195.3
		SD	0.004578	0.009273	0.012966	0.0000
	60	Mean	0.996477	0.999869	0.011748	7230
		SD	0.003516	0.012295	0.011688	0.0000
	70	Mean	0.998251	1.012891	0.008993	8675.8
		SD	0.002023	0.031169	0.007952	0.0000

Table 5 Comparison of PSO-CATV with PSO-TVIW, PSO-TVAC and jDE part II

F	*Dim*	*Measures*	*PSO-TVIW*	*PSO-TVAC*	*PSO-CATV*	*jDE*
f8	50	Mean	21.115114	21.095453	21.119545	21.1465
		SD	0.048413	0.056330	0.042264	0.031819
	60	Mean	21.163687	21.165886	21.191519	21.1965
		SD	0.043717	0.044411	0.042322	0.0261
	70	Mean	21.200285	21.198274	21.231040	21.224
		SD	0.048999	0.040314	0.048431	0.04808
f9	50	Mean	291.899708	287.740901	186.063673	0.0000
		SD	28.896009	26.897532	209.810298	0.0000
	60	Mean	367.496346	360.830206	684.778702	0.0000
		SD	35.602923	33.208458	527.462884	0.0000
	70	Mean	430.257996	434.277740	1,236.117304	0.0000
		SD	37.392734	33.017474	455.333121	0.0000
f10	50	Mean	508.576562	490.925751	469.395445	113.425
		SD	241.029088	66.944758	76.089387	7.035712473
	60	Mean	1,251.609138	605.504681	557.828100	134.815
		SD	619.303092	67.921327	82.476083	7.7428
	70	Mean	1,900.690573	706.790286	683.369855	188.05
		SD	573.614498	85.963727	91.500468	5.62856
f11	50	Mean	84.356621	34.680733	33.995479	52.0355
		SD	6.656662	6.215943	5.392729	14.68024
	60	Mean	104.211198	45.519844	42.041221	60.6085
		SD	2.621380	6.917720	6.358932	20.4007
	70	Mean	122.432169	53.992128	51.324729	72.847
		SD	8.117650	7.914189	3.887204	27.08218
f12	50	Mean	64,292.926	64,548.556	3,435,963.583	36695
		SD	29,754.286	34,908.068	4,708,418.529	37,611.00969
	60	Mean	78,812.533	87,337.496	12,242,957.869	49,456.5
		SD	33,183.514	48,051.435	5,935,467.345	1,642.60905
	70	Mean	127,746.286	123,025.408	18,825,897.822	50,784.5
		SD	53,646.227	56,343.148	5,704,889.387	26,141.0305
f13	50	Mean	243.747263	33.841916	10.225835	11.293
		SD	951.548735	4.519735	0.02174919	0.0028284
	60	Mean	2,212.361500	42.905302	46.441168	16.467
		SD	3,131.156487	7.183050	6.974479	0.32526
	70	Mean	6,565.684216	52.922639	59.530015	20.859
		SD	3,543.521298	6.606150	8.653405	2.070408
f14	50	Mean	21.83099	21.681058	21.757713	22.6245
		SD	0.529896	0.626972	0.444706	0.099702
	60	Mean	26.662176	26.813936	26.210791	27.3485
		SD	0.559658	0.589835	0.700757	0.089802
	70	Mean	31.400545	31.008263	31.735995	32.335
		SD	0.516103	0.875929	0.462124	0.1385

Fourth component in equation (12) represents cognitive avoidance and is considered as negative since it represents distraction, which is opposite to cognitive acceleration and social acceleration. To control the effect of this avoidance on a particle a cognitive avoidance coefficient C_3 is used along with randomness R_3 in range [0, 1]. Position equation remains unaltered as in equation (2). This new addition to the existing PSO is referred as PSO with cognitive avoidance (PSOCA). Effect of the newly added component is shown with an example in Figure 1. Black circle represents current position of a particle. Black diamond represents next position of the particle guided by pbest and gbest only. White diamond represents next position influenced by cognitive avoidance component. Proposed approach avoids movement towards worst solution and pushes final solution towards either pbest or gbest. In this case, it moves towards pbest.

A population of particles is initialised with randomly generated positions and velocities. Fitness of each particle is evaluated with user defined objective function. At each generation velocity of the particle is updated with equation (12) and next positions of particles are evaluated with equation (2). At each generation a particle finds best position or worst position, notes down that position, and updates its current pbest, pworst and gbest. Generally, velocities of particles are controlled with predefined values. If any particle gains larger velocity than the predefined velocity, modulus of the velocity is considered for updating positions. We have not considered any predefined velocity limits for particles. Let particles move with any finite velocity outside the search space then that movement is controlled with defined limits of each dimensions.

Although, strategically, we have overcome the problem of unfruitful moves, the problem of misguidance still remains in PSOCA. As the method incorporates probabilistic move by avoiding unfruitful moves, which may again misguide particles in some cases where optimal value is nearby the present pworst value. Particle may wrongly avoid move towards the optimal solution and may never reach the optimal solution. There has to be some mechanism so that such misinterpretation can compensated in successive iterations. To overcome such situation a varying cognitive avoidance coefficient (C_3) is introduced to PSOCA. A very similar mechanism used for varying weight in PSO-TVIW is utilised here for varying C_3. We refer this extension to PSOCA as time varying PSOCA (PSO-CATV). C_3 is updated in each iteration as follows:

$$C_3 = C_{3\min} + \left(C_{3\max} - C_{3\min}\right)\left(\frac{mIter - Iter}{mIter}\right) \quad (13)$$

Here, *mIter* is the maximum number of iterations and *Iter* is the current iteration number. It is clear from the equation (13) that the value of C_3 gradually decreases with successive iterations. Hence, effect of cognitive avoidance reduces as iteration increases. So, even if situation arises where solution are nearby the pworst, particle can reach the optimal solution as effect of misinterpretation reduces.

5 Experimental evaluation

5.1 Benchmark functions and performance metrics

For analysing results of the proposed approach, two sets of benchmark functions are used. First set comprises very popular four benchmark functions that were used for performance evaluation. Among these Rosenbrock's function is uni-modal, where as Ackley's, Rastrigin's and Schwefel's functions are multi-modal function. All functions have global optimal solution at or near the origin except Schwefel's function, which have global optimal solution at the edges of the solution domain. Function definitions and their initial ranges are shown in Table 1 and Table 2, respectively.

Second set comprises 25 benchmark functions that were presented in CEC 2005 special session.

- Five unimodal functions
  - a f1: Shifted sphere function.
  - b f2: Shifted Schwefel's problem 1.2.
  - c f3: Shifted rotated high conditioned elliptic function.
  - d f4: Shifted Schwefel's problem 1.2 with noise in fitness.
  - e f5: Schwefel's problem 2.6 with global optimum on bounds.
- 20 multimodal functions
  - a seven basic functions
    - 1 f6: Shifted Rosenbrock's function.
    - 2 f7: Shifted rotated Griewank function without bounds.
    - 3 f8: Shifted rotated Ackley's function with global optimum on bounds.
    - 4 f9: Shifted Rastrigin's function.
    - 5 f10: Shifted rotated Rastrigin's function.
    - 6 f11: Shifted rotated Weierstrass function.
    - 7 f12: Schwefel's problem 2.13.
  - b two expanded functions
    - 1 f13: Expanded extended Griewank's plus Rosenbrock's (Ef8f2)
    - 2 f14: Shifted rotated expanded Scaffers F6.
  - c 11 hybrid composition functions
    - 1 f15: Hybrid composition function 1
    - 2 f16: Rotated hybrid composition function 1
    - 3 f17: Rotated hybrid composition function 1 with noise in fitness
    - 4 f18: Rotated hybrid composition function 2
    - 5 f19: Rotated hybrid composition function 2 with a narrow basin for the global optimum
    - 6 f20: Rotated hybrid composition function 2 with the global optimum on the bounds
    - 7 f21: Rotated hybrid composition function 3

8 f22: Rotated hybrid composition function 3 with high condition number matrix

9 f23: Non-continuous rotated hybrid composition function 3

10 f24: Rotated hybrid composition function 4

11 f25: Rotated hybrid composition function 4 without bounds.

Optima of all the functions have been displaced from the origin or from the previous position to ensure that optimal solutions can never be obtained in centre of the domain. This displacement mechanism has made it difficult for the algorithms which have central tendency.

Since cognitive avoidance approach is added to PSO to improve overall performance, so proposed PSOCA is compared with standard PSO only. However, improved version of PSOCA i.e. PSO-CATV is compared with other two variants of PSO, PSO-TVIW and PSO-TVAC. Along with these two variants of PSO, we have also considered another state-of-the-art competitor jDE (Brest et al., 2006) to compare performance of PSO-CATV. This is an extension to the differential evolution approach. jDE can adapt parameters CR and F, suitable to corresponding situation.

5.2 *Analysis methods*

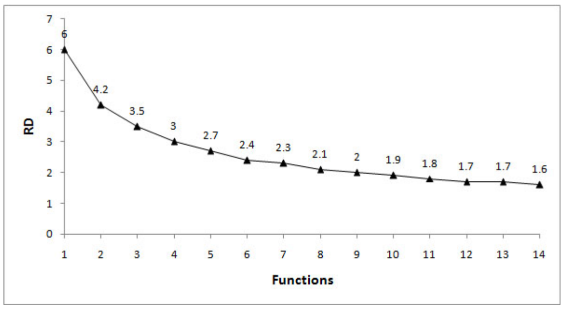
To analyse results obtained, we have considered two statistical approaches, value-based approach and rank-based approach. In value-based approach, we have evaluated two performance metrics mean and standard deviation. These metrics are evaluated for each algorithm with different functions from benchmark function sets in three different dimensions. In rank-based approach, we have adopted chess rating system for evolutionary algorithms (CRS4EAs) (Veček et al., 2014) to compare and rank PSO-CATV with respect to other competitor. Apart from this statistical approach, we also have considered graphical approach to visualise and compare convergence of PSO-CATV with other. We have noted best values obtained at each generation for different functions to visualise convergence and relative exploration or exploitation.

5.3 *Experimental setup*

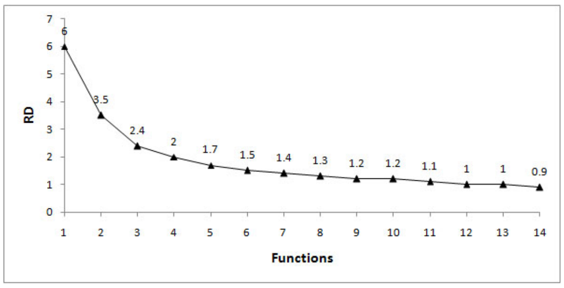
5.3.1 *Environment settings*

First set of benchmark functions are tested with dimensions 10, 20 and 30 to evaluate PSOCA. Population size is considered for this case is 40 and maximum iteration limit as 1000 for each run to made comparisons more precise. Second set 25 functions CEC 2005 special session is divided in two subsets. Subset 1 includes first 14 functions and subset 2 includes remaining 11 hybridised functions. Subset 1 is considered for both statistical analysis methods to evaluate PSO-CATV. For both value-based method and CRS4EAs considered dimensions are 50, 60 and 70. As increase in dimensionality increases complexity of the problem we have considered larger population size 100 for this case. Also considered different maximum iteration limit depending on there complexity level. For dimensions 50, 60 and 70, maximum iteration or generation limit are considered as 3,000, 4,000 and 5,000, respectively. For all statistical analysis, each function and corresponding considered dimension are executed over 50 trials to present performance metrics. CRS4EAs is utilised in two ways to rank algorithms. First, ranking is done by considering single dimension at a time and second ranking is done by considering all dimension at the same time. The initial CRS4EAs parameter settings for any new algorithm is considered as follows: rating R = 1,500, rating deviation RD = 350 and rating volatility σ = 0.06. The 99.7% confidence interval RI is used for analysis. This RI indicates that the difference between two ratings is significant if the difference is larger than 3 × RD. Code for CRS4EAs is obtained from Github (2013).

Figure 2 Change in rating deviation (RD) after addition of one function



(a)



(b)

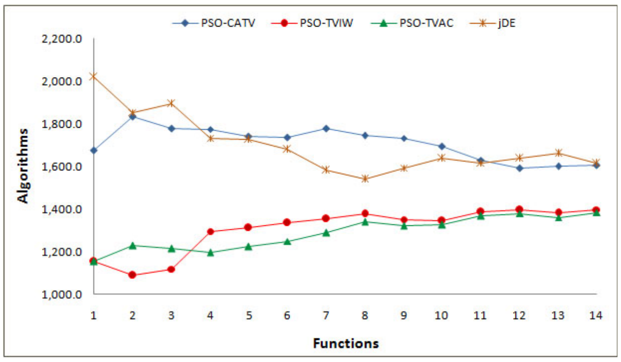
Notes: For all single dimension ranking RD changes at similar rate as shown in (a). Change in RD for ranking of algorithms with all dimensions taken at same time is shown in (b).

Table 6 Ranking interval (RI) of algorithms

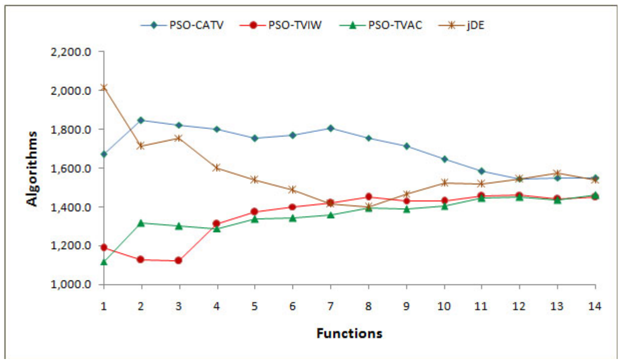
Algorithms	*Dim 50*	*Dim 60*	*Dim 70*	*All Dim*
PSO-CATV	[1,599, 1,608.6]	[1,542.9, 1,552.5]	[1,614.6, 1,624.2]	[1,587.6, 1,593]
jDE	[1,609.2, 1,618.8]	[1,534.7, 1,544.3]	[1,548.5, 1,558.1]	[1,566.2, 1,571.6]
PSO-TVAC	[1,379.7, 1,389.3]	[1,456.3, 1,465.9]	[1,437.7, 1,447.3]	[1,426.6, 1,432]
PSO-TVIW	[1,392.9, 1,402.5]	[1,447, 1,456.6]	[1,380.1, 1,389.7]	[1,408.8, 1,414.2]

Note: Rating deviation (RD) for all single dimension was reached value 1.6 and for all dimension at same time reached value 0.9.

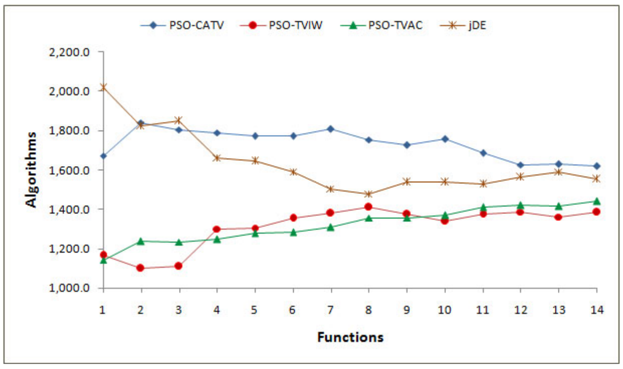
Figure 3 Change in rank of each algorithm after addition of one function to the tournament (see online version for colours)



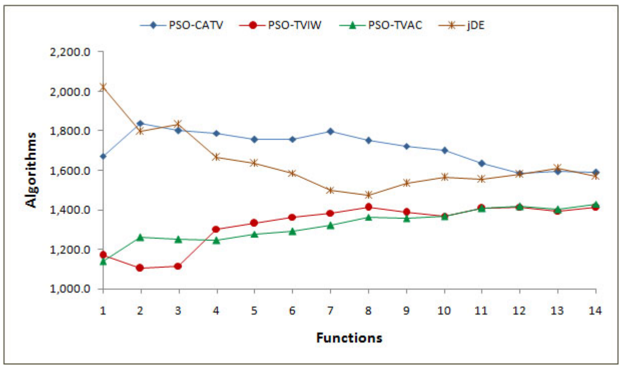
(a)



(b)



(c)



(d)

Note: Change in rank for single dimension ranking with dimension 50, 60 and 70 are presented in (a), (b) and (c) respectively. (d) Shows the same for all dimensions taken together.

For convergence analysis consider subset 2 of CEC 2005 benchmark functions. As subset 2 consist of 11 functions, all are hybrid functions of multiple functions of subset 1. These functions are more complex than subset 1. So, we have considered these functions to visualise how first algorithm reaches to optimal value. We keep population size 100 as before, but consider only one trial and observe best values over 1,000 generations. For all experiments population is initialised with uniformly distributed random values covering solution domain.

5.3.2 *Parameter settings*

Acceleration coefficients of PSOCA C_1, C_2 and newly added C_3 kept as constant values 0.6, 1.5 and 0.4, respectively. However, for PSO-CATV acceleration coefficients C_1 and C_2 are kept same constant value as PSOCA, but the newly added C_3 varied linearly from $C_{3\max} = 2.0$ to $C_{3\min} = 0.25$. For PSO-TIVW, the value of ω updated with $\omega_{\min} = 0.4$ and $\omega_{\max} = 0.9$ (Shi and Eberhart, 1999). For PSO-TVAC, value of C_1 varied from 2.5 to 0.5, C_2 from 0.5 to 2.5 and ω from 0.9 to 0.4 (Ratnaweera et al., 2004). jDE not required any parameter setting as its parameters CR and F are adapted during execution.

5.4 *Result analysis*

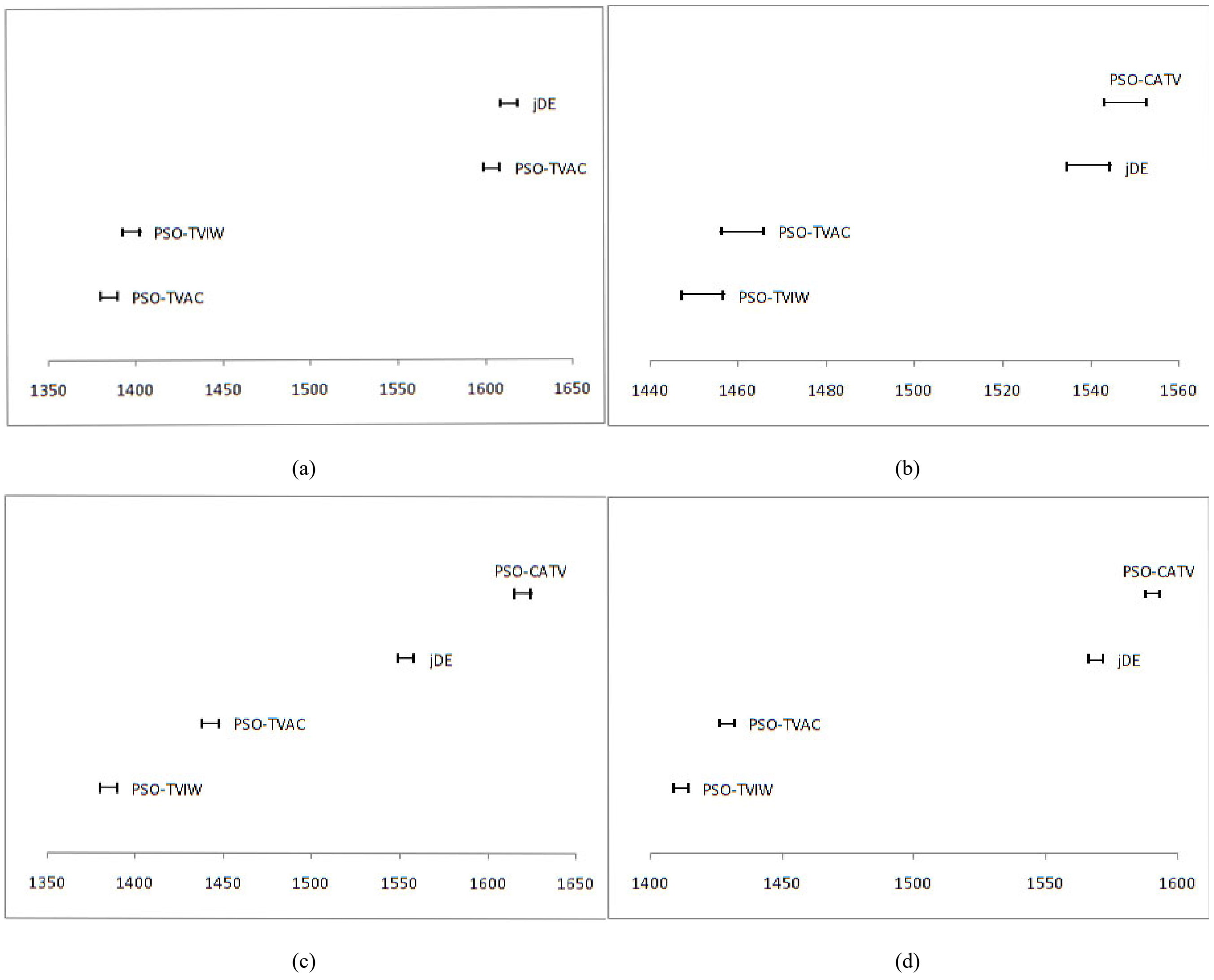
5.4.1 *Value-based analysis*

Results of PSOCA and PSO on the first set of benchmark functions, performance metrics of optimal solutions over 50 trials are presented as in Table 3. For Rastrigin's function in all dimensions PSOCA performs better than PSO in terms of both mean and standard deviation. For Ackley's function in dimensions 10 and 20 shows almost similar results, but in 30 PSOCA performs better than PSO. In dimension 10 both reaches optimal value. For Rosenbrock's function in dimension 10 performance of PSOCA is poor. In dimension 20, PSOCA performs very well. In dimension 30 shows little improvement in performance than PSO. From these experiments, it is clear that introduction of cognitive avoidance to PSO not only improves results but also gives benefit for higher dimensional problems. Performance of PSOCA seems better with increment of dimensionality.

Results of PSO-CATV and other competitor on first seven of subset 1 of CEC 2005 special session benchmark functions with the performance metrics of optimal solutions over 50 trials are presented as in Table 4. This is clear from the results presented in Table 4 that PSO-CATV outperforms over PSO-TVIW, PSO-TVAC and jDE in functions f1 and f2 for all dimensions. In f3, PSO-CATV shows better result than PSO-TVIW and PSO-TVAC, and shows very similar results to jDE. In f4, for all dimensions PSO-CATV outperforms jDE and PSO-TVAC, but fall behind PSO-TVIW. In f5, for dimensions 50 and 60 performance of PSO-CATV is almost similar to both the competitor, but for dimension 70 PSO-CATV shows drastic improvement. However, for dimension 50 performance of jDE is better than PSO-CATV and it gives a better result than jDE on dimensions 60 and 70. Again in functions f6 and f7, PSO-CATV outperforms over PSO-TVIW and PSO-TVAC for all dimensions. But, in f6, jDE performs better than PSO-CATV. Performance of PSO-CATV again seems better than jDE and noted that jDE is getting stick to some specific values over all 50 trials. Which indicates that jDE might have fall into that local optima and unable to come out of that optima.

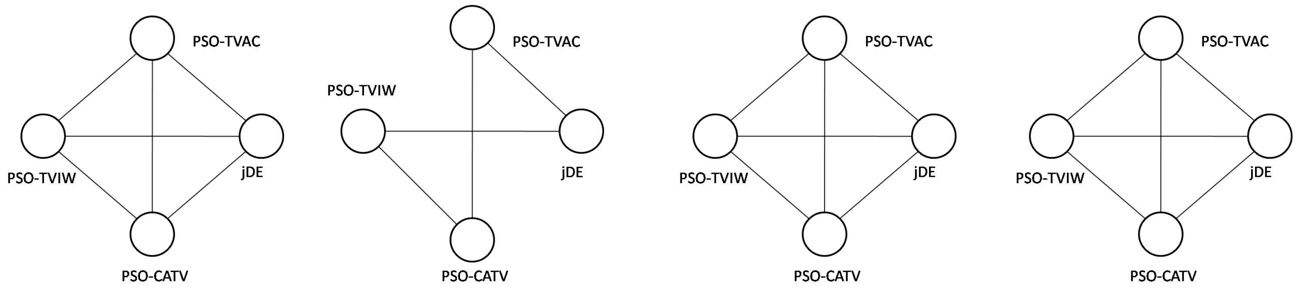
Results on rest of the functions of subset 1 are presented in Table 5. In f8, performance of PSO-CATV shows very similar performance as jDE, PSO-TVIW and PSO-TVAC. In f9, performance of PSO-CATV is poor. In f10, PSO-CATV outperforms in almost all dimensions than PSO-TVIW and PSO-TVAC, but poorer than jDE. Again in f11, PSO-CATV outperforms all three competitors in almost all dimensions. In f12, PSO-CATV shows very poor performance. However, all three competitors also show poor result in this function, though comparatively better than PSO-CATV. In f13, PSO-CATV outperforms PSO-TVIW but, falls behind PSO-TVAC and jDE. In f14, performance of PSO-CATV is almost similar but comparatively better than all three competitors. Overall performance of PSO-CATV is better than all three PSO-TVIW, PSO-TVAC and jDE in all functions except f9 and f12. Our observation regarding cognitive avoidance for PSOCA remains for PSO-CATV as well.

Figure 4 99.7% confidence intervals for the algorithms ratings from Table 7



Notes: Here, X-axis represents ratings of algorithms'. (a), (b) and (c) presents 99.7% confidence interval corresponding to single dimensional ranking of dimensions 50, 60 and 70, respectively. (d) Presents the same for all dimensions taken together. Only in (b), i.e., for dimension 60 has overlapping interval with PSO-TVIW, PSO-TVAC and PSO-CATV, jDE.

Figure 5 Significance difference detected with CRS4EAs



(a) (b) (c) (d)

Notes: Two algorithms are significantly different when 99.7% confidence interval do not overlap. Significance difference detected with CRS4EAs for dimensions 50, 60, 70 and all at the same time is shown in (a), (b), (c) and (d), respectively.

Figure 6 Convergence comparison of PSO-CATV with PSO-TVIW, PSO-TVAC and jDE

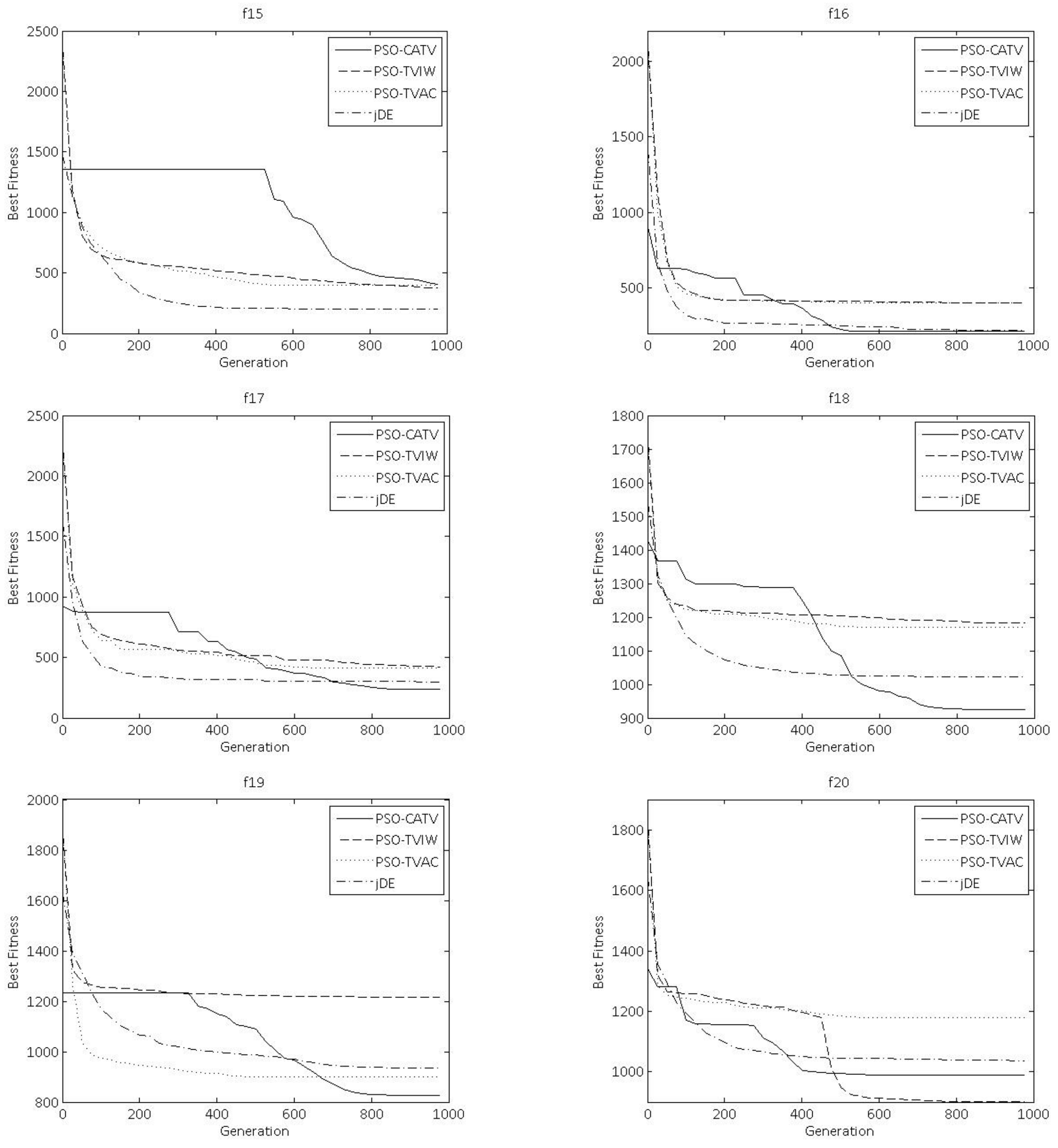
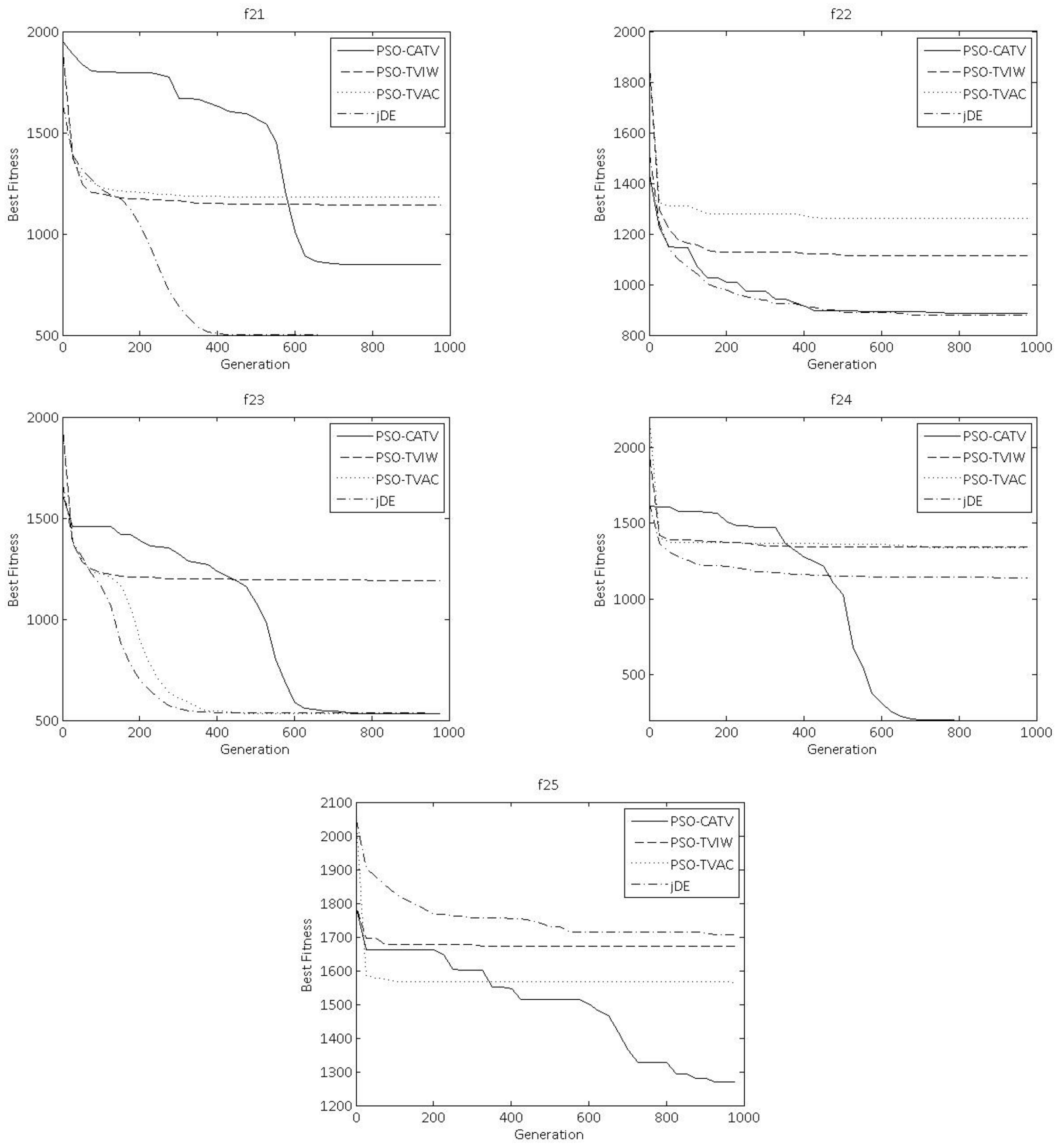


Figure 6 Convergence comparison of PSO-CATV with PSO-TVIW, PSO-TVAC and jDE (continued)



5.4.2 Rank-based analysis

Results obtained with CRS4EAs are presented in Table 7, where algorithms' ranking corresponding to different dimensions are shown. Hence, we have four rankings corresponding to the dimensions 50, 60, 70 and all (i.e., 50, 60 and 70 taken same time in the tournament). An algorithm's ranking R represents the power of the algorithm after all executed tournaments. Along with this ranking, we have also considered measurements like RD and rating interval (RI) as suggested by the Glicko-2 rating system with known explanations (Veček et al., 2014). We have noted change in ranking with addition of a function. We have added functions in sequence of f1, f2, … up to f14 in the tournament to do so. Change in ranking of algorithms with such incremental tournament is presented in Figure 3. With the change in ranking, we have also noted change in RD with addition of functions to the tournament. We have observed same change in RD for single dimensions, but different for all dimensions together. Change in RD for single and all dimensions are shown in Figure 2. At the end of tournament RD values for single dimension and all dimensions reached to 1.6 and 0.9, respectively. Table 7 displays final ranking after the end of the tournaments. 99.7% confidence interval [R-3 × RD, R+3 × RD] is considered to make sure that any possible error is minimal. RI for all algorithms with respect to their final RD values of ranking is shown in Table 6. The leaderboards for all four ranking are designed with the best algorithm on the top with a rank of 1. A graphical presentation of these results are presented in Figure 5. Nodes in the graph present algorithms. Two nodes are connected if their RIs do not overlap. Two connected algorithms indicate that these two algorithms are significantly different.

Table 7 Ranking of algorithms

Algorithms	*Dim 50*	*Dim 60*	*Dim 70*	*All Dim*
PSO-CATV	1,603.8	1,547.7	1,619.4	1,590.3
jDE	1,614.0	1,539.5	1,553.3	1,568.9
PSO-TVAC	1,384.5	1,461.1	1,442.5	1,429.3
PSO-TVIW	1,397.7	1,451.8	1,384.9	1,411.5

Notes: Columns labelled with Dim 50, Dim 60 and Dim 70 shows rankings when for ranking considered single dimension 50, 60 and 70, respectively. Columns labelled with all Dim shows rankings when all three dimensions are taken together for ranking.

Rankings obtained with CRS4EAs shows clearly superiority over other three competitors as in Table 7. In 3 of 4 rankings except for dimension 50 PSO-CATV shows highest ranks. In 50 dimension also, PSO-CATV shows higher ranking than PSO-TVIW and PSO-TVAC. Though PSO-CATV lagging behind jDE in final rank on dimension 50, it would be clear from change in rankings with addition of new function (as shown in Figure 3) to the tournament that most of the cases PSO-CATV had higher ranking than jDE. But, as PSO-CATV performs badly in f12 its ranking degrades and it was also noted during value-based analysis. As mentioned in Veček et al. (2014), a skewed result can effect very badly in ranking of algorithms, but still with such skewness in results PSO-CATV managed to have highest ranking for other cases. Normally, addition of new function causes decrement in RD value (as shown in Figure 2), but after f12 it remains same indicating something odd things present in results. Again from view point of RI (as presented in Table 6) and ranking, it is clear that PSO-CATV has been the best among these competitors. Ranking shows superiority and RI indicates significance level of that superiority. As mentioned in Veček et al. (2014), non-overlapping interval means significant difference in algorithms. RI overlapping can be clearly visible in leaderboards presented in Figure 4. Graphical presentation of these results presented in Figure 5 shows that only in case dimension 60 have overlapping. Hence, rest of the ranking indicates significant difference in algorithms performance.

5.4.3 Convergence analysis

Convergence results on subset 2 of CEC 2005 special session benchmark functions that includes f15 to f25 hybridised functions are displayed in Figure 6. PSO-CATV shows slow convergence at beginning as it should be, because initially PSO-CATV explores and as gradually decreases cognitive avoidance component it start exploiting. Hence, convergence of PSO-CATV is observable almost after 50% of generations completed. While exploration for all three components seems to be ends early. Though they exploits thereafter, best fitness value does not improves significantly and after a while remains constant. It is also notable that despite being PSO-CATV starts with slow convergence, but most of the cases at the end of execution gradually improves best fitness and results better than all three competitors. Though in functions f15, f20 and f21 show exception to this observation as PSO-CATV lagging behind some these competitors, exploration of solution space is observable. PSO-CATV shows extensively very high convergence rate in f24 and f25, it leaves all three competitors far behind both in terms of best result and exploration. Overall performance of PSO-CATV seems to be better than all three competitors.

6 Conclusions

This paper introduces a novel mechanism to improve performance of PSO by avoiding particle's unfruitful moves. New additional component called cognitive avoidance is introduced to the velocity equation of PSO. Cognitive avoidance coefficient controls the effect of this component. The approach tracks personal worst positions and directs particle to avoid movement towards such positions. The concept of linearly decreasing cognitive avoidance coefficient is also introduced to negotiate the negative effect and to maintain balance between exploration and exploitation. Proposed approach is evaluated with two sets of benchmark functions comprising total of 29 functions. Empirical result shows performances of two proposed cognitive avoidance schemes PSOCA and PSO-CATV are better than three previous versions of PSO and state-of-the-art competitor jDE. Result is evaluated with three different approaches, value-based, rank-based and convergence-based. In value-based approach, PSO-CATV shows higher quality solutions than all competitors and is also ranked higher. Gradually improved convergence of PSO-CATV is also better than other competitors, as most of the cases it able to reach nearer to the optimal value. Hence, it is worthwhile to mention that additional cognitive avoidance component definitely improves PSO to certain extent.

References

Arya, M., Deep, K. and Bansal, J.C. (2014) 'A nature inspired adaptive inertia weight in particle swarm optimization', *International Journal of Artificial Intelligence and Soft Computing*, Vol. 4, Nos. 2/3, pp.228–248.

Biswas, A., Kumar, A. and Mishra, K.K. (2013) 'Particle swarm optimization with cognitive avoidance component', *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp.149–154.

Brest, J., Greiner, S., Bokovi, B., Mernik, M. and Umer, V. (2006) 'Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems', *IEEE Trans. on Evolutionary Computation*, Vol. 10, No. 6, pp.646–657.

Chen, W., Xiao, R. and Lu, H. (2014) 'A chaotic PSO approach to multi-mode resource-constraint project scheduling with uncertainty', *International Journal of Computational Science and Engineering*, Vol. 6, Nos. 1/2, pp.5–15.

Chuang, L., Chang, H., Tu, C. and Yang, C. (2008) 'Improved binary PSO for feature selection using gene expression data', *Computational Biology and Chemistry*, Vol. 32, No. 1, pp.29–38.

Clerc, M. and Kennedy, J. (2002) 'The particle swarm-explosion, stability, and convergence in a multidimensional complex space', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp.58–73.

Črepinšek, M., Liu, S.H. and Mernik, M. (2013) 'Exploration and exploitation in evolutionary algorithms: a survey', *ACM Computing Surveys*, Vol. 45, No. 3, pp.1–35.

Eberhart, R.C. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', *Proceedings of Sixth International Symposium on Micro Machine and Human Science*, pp.39–43.

Eberhart, R.C. and Shi, Y. (2001) 'Tracking and optimizing dynamic systems with particle swarms', *Proceedings of IEEE International Congress on Evolutionary Computation*, pp.94–100.

Eberhart, R.C., Dobbins, R.W. and Simpson, P. (1996) *Computational Intelligence PC Tools*, Academic Press, Boston.

Evolutionary Algorithms Rating System (Github) (2013) (version 1) [online] https://github.com/matejxxx/EARS (accessed April 2014).

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Goldberg, D. (1990) 'A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing', *TCGA 90003*, Engineering Mechanics, University of Alabama.

Gong, T. and Tuson, A.L. (2007) 'Binary particle swarm optimization: a formal analysis approach', *Proceeding: GECCO '07 Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*, ACM New York, NY, USA.

Holland, J. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Honghao, C., Zuren, F. and Zhigang, R. (2013) 'Community detection using ant colony optimization', *2013 IEEE Congress on Evolutionary Computation (CEC)*, 20–23 June, pp.3072–3078.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proceedings of IEEE International Conference on Neural Networks*, pp.1942–1948.

Kennedy, J., Eberhart, R.C. and Shi, Y. (2001) *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco.

Khosla, M., Sarin, R.K. and Uddin, M. (2014) 'Evolutionary design of efficient type-2 fuzzy models from noisy data using hybrid PSO model', *International Journal of Swarm Intelligence*, Vol. 1, No. 2, pp.156–178.

Liu, Y., Yao, M. and Zhu, R. (2011) 'A novel hybrid model for image classification', *International Journal of Computational Science and Engineering*, Vol. 6, Nos. 1/2, pp.96–104.

Mathematics Statistics Research Group, Chinese Academy of Science (MSR Group) (1975) *Orthogonal Design*, in Chinese, People Education Publication, Beijing, China.

Mendes, R. (2004) *Population Topologies and their Influence in Particle Swarm Performance*, PhD dissertation, Universidade do Minho.

Montgomery, D.C. (2000) *Design and Analysis of Experiments*, 5th ed., Wiley, New York.

Parsopoulos, K. and Vrahatis, M. (2002) 'Recent approaches to global optimization problems through particle swarm optimization', *Natural Computing*, Vol. 1, pp.235–306.

Rajagopalan, N. and Mala, C. (2014) 'Channel allocation scheme for cellular networks using evolutionary computing', *International Journal of Artificial Intelligence and Soft Computing*, Vol. 4, Nos. 2/3, pp.212–227.

Ratnaweera, A., Halgamuge, S. and Watson, H.C. (2004) 'Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.240–255.

Sharma, H., Bansal, J.C. and Arya, K.V. (2014) 'Power law-based local search in artificial bee colony', *International Journal of Artificial Intelligence and Soft Computing*, Vol. 4, Nos. 2/3, pp.164–194.

Shayeghi, H., Mahdavi, M. and Bagheri, A. (2010) 'Discrete PSO algorithm based optimization of transmission lines loading in TNEP problem', *Energy Conversion and Management*, Vol. 51, No. 1, pp.112–121.

Shi, Y. and Eberhart, R.C. (1999) 'Empirical study of particle swarm optimization', *Proceedings of IEEE International Congress on Evolutionary Computation*, Vol. 3, pp.101–106.

Srinivas, M. and Patnaik, L. (1994) 'Adaptive probabilities of crossover and mutation in genetic algorithms', *IEEE Transactions on System, Man and Cybernetics*, Vol. 24, No. 4, pp.656–667.

Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Auger, A. and Tiwari, S. (2005) *Problem Definitions and Evaluation Criteria for the CEC2005 Special Session on Real Parameter Optimization*, Nanyang Technological University, Technical Report.

Tasgetiren, M.F. and Lian, Y. (2004) 'A binary particle swarm optimization algorithm for lot sizing problem', *Journal of Economic and Social Research*, Vol. 5, No. 2, pp.1–20.

Veček, N., Mernik, M. and Črepinšek, M. (2014) 'A chess rating system for evolutionary algorithms: a new method for the comparison and ranking of evolutionary algorithms', *Information Sciences*, 1 September, Vol. 277, pp.656–679.

Zhan, Z.H. and Zhang, J. (2011) 'Orthogonal learning particle swarm optimization for power electronic circuit optimization with free search range', *IEEE Congress on Evolutionary Computation (CEC)*, pp.2563–2570.

Zhan, Z.H., Zhang, J., Li, Y., Shi, Y. (2011) 'Orthogonal learning particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 6, pp.832–847

Zhang, L., Lu, J., Chen, L. and Zhang, J. (2013) 'Particle swarm optimisation algorithm for radio frequency identification network topology optimisation', *International Journal of Computational Science and Engineering*, Vol. 6, Nos. 1/2, pp.16–23.