Int. J. Advance Soft Compu. Appl, Vol. 8, No. 2, July 2016
ISSN 2074-8523

Improved Centripetal Accelerated Particle Swarm Optimization

Zahra Beheshti[1,2], Siti Mariyam Shamsuddin[1,2] , Shafaatunnur Hasan[1,2] and Nur Eiliyah Wong [1,2]

[1]UTM Big Data Centre,
Ibnu Sina Institute for Scientific and Industrial Research,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

[2]Faculty of Computing
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
e-mail: bzahra2@live.utm.my; mariyam, shafaatunnur, nureiliyah@utm.my

Abstract

Particle swarm optimization (PSO) is a bio-inspired optimization algorithm that imitates the social behavior of bird flocking, fish schooling and swarm theory. Although PSO has a simple concept and is easy to implement, it may converge prematurely because of its poor exploration when solving complex multimodal problems. Centripetal accelerated particle swarm optimization (CAPSO) is an enhanced particle swarm optimization (PSO) scheme combined with Newton's laws of motion. It has shown an effective algorithm in solving optimization problems however; its performance can be enhanced similar to other evolutionary computation algorithms. In this study, an improved CAPSO (ICAPSO) and improved binary CAPSO (IBCAPSO) are introduced to accelerate the learning procedure and convergence rate of optimization problems in the real and binary search spaces. The proposed algorithms are evaluated by twenty high-dimensional complex benchmark functions. The results showed that the methods substantially enhance the performance of the CAPSO for both the real and binary search spaces in terms of the convergence speed, global optimality, and solution accuracy.

Keywords: *Accelerated Particle Swarm Optimization, Centripetal, Optimization, Global and Local optimum, Global and Local Topology, Particle Swarm Optimization.*

1 Introduction

PSO is an optimization algorithm introduced by Kennedy and Eberhart [1, 2]. It has extensively applied in various fields [3-7] because of simple concepts, ease of implementation and having few parameters. However, it suffers from a premature convergence rate and trapping into local optimum when solving complex multimodal problems [8-11]. Many variant PSO algorithms have been proposed to overcome these drawbacks. The majority of methods improved the performance of PSO through parameter selection [12, 13], swarm topology [14, 15, 16, 17], and hybridization PSO with the other techniques such as mutation [18], crossover [19, 20], orthogonal learning strategy [21, 22], elitist learning strategy [8], chaos [23], and so on.

LPSO (ring topological structure PSO) [14] and VPSO (Von neumann topological structure PSO) [15] were introduced by Kennedy and Mendes. Parsopoulos and Vrahatis [16] proposed unified particle swarm optimization (UPSO) that combined the global and local topologies in PSO. Fully informed particle swarm (FIPS) algorithm is suggested by Mendes et al. [24]. The algorithm applied the information from whole neighborhoods to guide particles for finding the best solution. Liang and Suganthan suggested dynamic multi-swarm PSO (DMS-PSO) [25] in order to dynamically enhance the topological structure. A non-parametric particle swarm optimization (NP-PSO) was suggested by Beheshti et al. to solve global optimization problems [26]. HPSO-TVAC algorithm was introduced by Ratnaweera et al. [13], which used linearly time-varying acceleration coefficients to improve the exploitation and exploration abilities. Zhan et al. [8] introduced adaptive particle swarm optimization (APSO). The algorithm employed a real-time evolutionary state estimation procedure and an elitist learning strategy to increase the performance of PSO. In another study, an adaptive fuzzy particle swarm optimization (AFPSO) [27] was suggested to utilize fuzzy inferences that adaptively adjust acceleration coefficients rather than fixed constants. Liang et al. [28] presented comprehensive learning particle swarm optimization (CLPSO) to avoid the local optima by encouraging every particle to learn the other particles' behaviour from different dimensions. Also, Zhan et al. [22] proposed the orthogonal learning particle swarm optimization (OLPSO) algorithm. The OL strategy guides a particle to discover useful information from its personal best position and from its neighborhood particles' best positions to move in better directions. A median-oriented particle swarm optimization (MPSO) [10] was proposed by Beheshti et al. to carry out a global search over entire search space with accelerating convergence speed and avoiding local optima.

Also, a binary version of PSO (BPSO) was proposed by Kennedy and Eberhart [29] for discrete optimization problems. BPSO may easily get trapped in local optima and cannot converge well. Hence, several improved BPSO algorithms have been presented to address the disadvantages of BPSO [30-34].

Recently, Beheshti and Shamsuddin introduced CAPSO and local topology CAPSO (LCAPSO) utilizing the laws of motion in mechanics and the PSO algorithm for solving optimization problem for the high-dimensional real search space [11]. Moreover, the binary version of algorithms, BCAPSO and LBCAPSO suggested for the discrete search space. The algorithms were compared with gravitational search algorithm (GSA), PSO, and several well-known PSO algorithms in the literature such as LPSO, HPSO-TVAC, VPSO, DMS-PSO, CLPSO, and APSO. The results showed that the proposed methods provided a good performance compared with the other algorithms. The methods are effective in solving optimization problems [35-38]. To enhance the performance of the algorithms, two common techniques are often applied: self-adaptation of the parameters in the algorithm and combination with different algorithms.

In this study, a new quadratic crossover operator is proposed to incorporate with CAPSO and BCAPSO for both global and local topology. The aim is to improve the convergence speed, global optimality, and solution accuracy of algorithms in the high-dimensional search spaces. Twenty high-dimensional benchmark functions [39, 40] are selected to evaluate the performance of proposed methods.

The remainder of this paper is organized as follows. In Section 2, a brief review of CAPSO, LCAPSO, BCAPSO and LBCAPSO is given. The details of the proposed methods are explained in Section 3. In Section 4, the experimental results on some well-known benchmark functions are provided, and conclusions are presented in Section 5.

2 Related Work

2.1 CAPSO in the real search space

CAPSO [11] is an improved PSO scheme combined with the laws of motion in mechanics. In the algorithm, every particle applies four specifications to obtain a solution in the search space. They are position, velocity, acceleration and centripetal acceleration. The position and velocity of the *ith* particle in a *n*-dimensional search space are defined as follows:

$$\vec{X}_i = (x_{i1}, x_{i2}, ..., x_{id}, ..., x_{in}) \quad for \quad i = 1,2,...,N . \tag{1}$$

$$\vec{V}_i = (v_{i1}, v_{i2}, ..., v_{id}, ..., v_{in}) \quad for \quad i = 1,2,...,N . \tag{2}$$

A personal best position for every particle, $\vec{Pi}$, and the best position explored by population, $\vec{P_g}$, are computed to update the velocity and position.

$$\vec{P_i} = (p_{i1}, p_{i2}, ..., p_{id}, ..., p_{in}) \quad for \quad i = 1,2,...,N . \tag{3}$$

$$\vec{Pg} = \left(p_{g1}, p_{g2}, \ldots, p_{gd}, \ldots, p_{gn}\right). \quad (4)$$

In the algorithm, two models for choosing $\vec{Pg}$ are considered: *gbest* and *lbest* models. *gbest* is known as global topology and *lbest* is local topology. In global topology, the velocity of each particle is updated by the best fitness obtained by swarm whereas in the local model, the velocity is modified by the best fitness selected from each particle's neighborhood.

The velocity and the position of every particle are initialized randomly and the next velocity is computed as follows:

$$v_{id}(t+1) = v_{id}(t) + a_{id}(t) + A_{id}(t), \quad (5)$$

where $v_{id}(t+1)$ and $v_{id}(t)$ are the next and current velocity respectively. $a_{id}(t)$ represents the acceleration and $A_{id}(t)$ is centripetal acceleration of the *ith* particle.

The acceleration and the centripetal acceleration for the *ith* particle are obtained by Eq. (6) and Eq. (7), respectively.

$$a_{id}(t) = rand \times (p_{id}(t) - x_{id}(t)) + rand \times \left(p_{gd}(t) - x_{id}(t)\right), \quad (6)$$

$$A_{id}(t) = E_i(t) \times rand \times (p_{id}(t) - p_{md}(t) - x_{id}(t)), \quad (7)$$

where *rand* is a random number with uniform distribution between 0 and 1. $x_{id}(t)$ represents the current position. $p_{md}(t)$ is the current median position of particles in the *dth* dimension and $E_i(t)$ is the acceleration coefficient given as follows:

$$e_i(t) = \frac{fit_i(t) - GWfit(t)}{Avgfit(t) - GWfit(t)}, \quad (8)$$

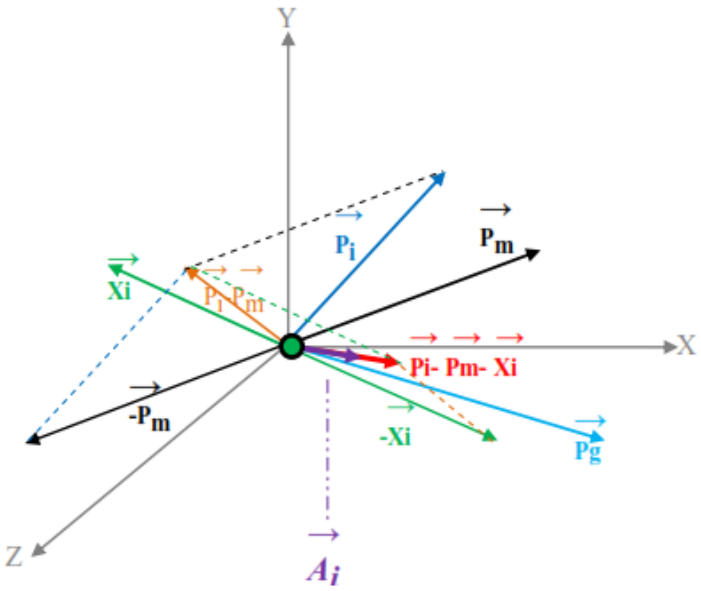
$$E_i(t) = \frac{e_i(t)}{\sum_{j=1}^{N} e_i(t)}, \quad (9)$$

where $fit_i(t)$ shows the fitness value of the particle *i*, $GWfit(t)$ represents the worst fitness value explored so far by the swarm. $Avgfit(t)$ is defined as follows:

$$Avgfit(t) = mean(fit_i(j)) \quad for \quad j = 1,2,\ldots,N. \quad (10)$$

Fig. 1 illustrates the graphical representation of $\vec{A_i}$ and $\vec{a_i}$. As shown in this Fig., $\vec{A_i}$ increases the convergence rate of algorithm and helps to escape from local optima.

(a)



(b)

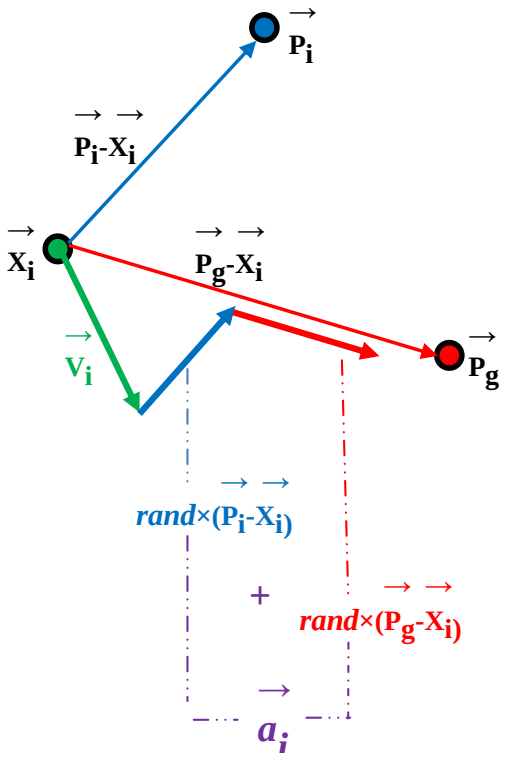


Fig. 1: Graphical representation of (a) $\vec{A_i}$ and (b) $\vec{a_i}$ [11]

The next position of *ith* particle is obtained based on the current position, acceleration and the next velocity, as follows:

$$x_{id}(t+1) = x_{id}(t) + \frac{1}{2} \times a_{id}(t) + v_{id}(t+1). \quad (11)$$

In PSO, the next velocity and position of particle i on dimension d are updated as follows:

$$v_{id}(t+1) = w \times v_{id}(t) + C_1 \times rand \times (p_{id}(t) - x_{id}(t)) + C_2 \times rand \times (p_{gd}(t) - x_{id}(t)), \quad (12)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \quad (13)$$

where C_1 and C_2 are acceleration coefficients and w is inertia weight.

2.2 BCAPSO in the binary search space

In the binary search space some modifications should be applied in the CAPSO algorithm. The concepts of BCAPSO are similar to the original CAPSO, but updating the position is switched between '0' and '1'. LBCAPSO is also the local

topology of BCAPSO. Therefore, Eq. (1) to Eq. (10) are used in BCAPSO and the next position of the particle for minimization problems is modified based on Eq. (14) to Eq. (16):

$$y_{id}(t+1)=\frac{1}{2}\times a_{id}(t)+v_{id}(t+1), \tag{14}$$

$$S(y_{id}(t+1))=\frac{0.5}{0.5+e^{-y_{id}(t+1)}}, \tag{15}$$

$$\begin{aligned} &if \quad rand<S(y_{id}(t+1)) \quad then \quad x_{id}(t+1)=1 \\ &else \quad x_{id}(t+1)=0 \quad for \quad i=1,2,\ldots,N \end{aligned}. \tag{16}$$

In Eq. (16), $x_{id}(t+1)$ changes to '1' if the random number is less than the given probability by the sigmoid function, otherwise, it modifies to '0'.

3 ICAPSO and IBCAPSO- proposed methods

To increase the convergence speed and solution accuracy in CAPSO and LCAPSO, a new quadratic crossover operator is proposed and incorporated in the algorithms. The operator is defined as Eq. (17). A new particle is created and if its fitness value is better than the best solution obtained by the swarm, it is replaced in the swarm.

$$\vec{\tilde{X}}=rand\times(\vec{X}_j-\vec{X}_k+\vec{P}_{gbest}), \tag{17}$$

where $\vec{P}_{gbest}$ is the best position obtained by the swarm, $\vec{X}_j$ and $\vec{X}_k$ are particles selected randomly from the swarm.

$\vec{P}_{gbest}$, $\vec{X}_j$ and $\vec{X}_k$ must be different from each other. If the position of $\vec{\tilde{X}}$ is better than the $\vec{P}_{gbest}$, the $\vec{P}_{gbest}$ is replaced by the new particle $\vec{\tilde{X}}$ in the swarm. In ICAPSO and ILCAPSO (Improved local topology CAPSO) algorithms, the Eq. (1) to Eq. (10) are applied as CAPSO and LCAPSO algorithms. LCAPSO updates the velocities and positions based on the local topology meanwhile, the particles' velocities and positions in ILCAPSO are modified based on the local topology and the global topology, respectively. If the position of $\vec{\tilde{X}}$ is better than the $\vec{P}_{gbest}$, the next position of the *ith* particle is changed as Eq. (19).

$$\tilde{a}_{id}(t)=rand\times(p_{id}(t)-x_{id}(t))+rand\times(\tilde{x}_d(t)-x_{id}(t)) \tag{18}$$

$$x_{id}(t+1)=x_{id}(t)+\frac{1}{2}\times\tilde{a}_{id}(t)+v_{id}(t+1). \quad (19)$$

At the beginning, the velocities and positions of particles are randomly initialized, and then each particle is evaluated based on its fitness value. $\vec{P}_i$ is computed for each particle and the best position explored by the swarm is selected as $\vec{P}_g$ ($\vec{P}_{gbest}$, for the *gbest* model and $\vec{P}_{lbest}$, for *lbest* model). The next particles' velocities will obtain based on the terms of $a_i(t)$, $E_i(t)$, $A_i(t)$, $\vec{P}_i$ and $\vec{P}_g$. If the position of $\vec{\tilde{X}}$ is better than the $\vec{P}_{gbest}$, the $\vec{P}_{gbest}$ is replaced by $\vec{\tilde{X}}$ in the swarm and the next position is computed as Eq. (19). Otherwise, $x_{id}(t+1)$ is calculated as Eq. (11). These steps will be continued until stopping criterion is met and the best solution is obtained by the algorithms. The procedure of the proposed algorithms for minimization problems is shown in Fig. 2.

In the binary search space, the particles' positions are initialized randomly with '0' or '1'. In IBCAPSO and ILBCAPSO (Improved local topology BCAPSO), the next positon is obtained in different way from BCAPSO and LBCAPSO. In the proposed methods in the binary search space, two positions are used: x_{id}^b and x_{id}. x_{id}^b has a binary value and x_{id} has a real value. The next velocity is computed as Eq. (5) that a_{id} and a_{id} are obtained as follows:

$$a_{id}(t)=rand\times\left(p_{id}(t)-x_{id}^b(t)\right)+rand\times\left(p_{gd}(t)-x_{id}^b(t)\right), \quad (20)$$

$$A_{id}(t)=E_{id}(t)\times rand\times\left(p_{id}(t)-p_{md}(t)-x_{id}^b(t)\right), \quad (21)$$

where x_{id}^b is the binary position of the *ith* particle.

```
Initialize the velocities and positions of population randomly.
while Terminating condition is not reached do
    for each particle i do
            if fit(X_i) < fit(P_i) then
                P_i = X_i
            end
```

if $fit(\vec{X}_i) < fit(\vec{P}_g)$ **then**

$\vec{P}_g = \vec{X}_i$ // P_g is P_{gbest} in global topology and P_{lbest} in local topology.

end

Update the next velocities of particles as Eq. (5).

Compute $\vec{\tilde{X}}$ as Eq. (17).

if $fit(\vec{\tilde{X}}) < fit(\vec{P}_{gbest})$ **then**

// In both global and local topologies, updating the next position is based
// on global topology

$\vec{P}_{gbest} = \vec{\tilde{X}}$

Update the next positions of particles as Eq. (19).

else

Update the next positions of particles as Eq. (11).

end

end // end for

end // end while

Fig. 2: General structure of the ICAPSO and ILCAPSO algorithms

Also, the real position of each particle, x_{id}, is computed as follows:

$$x_{id}(t+1) = x_{id}^{b}(t) + \frac{1}{2} \times a_{id}(t) + v_{id}(t+1) \tag{22}$$

This position, $x_{id}(t+1)$, is transferred into a sigmoid function (Eq. (23)) in the interval [0, 1]. Then, a new value (xx'_{id}) of '0' or '1' is obtained as Eq. (24).

$$S(x_{id}(t+1)) = \frac{0.5}{0.5 + e^{-x_{id}(t+1)}}, \tag{23}$$

$$\begin{array}{l} if \quad rand < S(x_{id}(t+1)) \quad then \quad xx'_{id} = 1 \\ else \quad xx'_{id} = 0 \quad for \quad i = 1,2,\ldots,N \end{array}, \tag{24}$$

The next binary position, $x_{id}^{b}(t+1)$, is updated if the fitness value of xx'_{id} is better than the fitness value of previous position $\left(x_{id}^{b}(t)\right)$, otherwise the next position is equal to the previous binary position as follows:

$$\begin{aligned} &if \quad fit(xx'_{id}) < fit\left(x^b_{id}(t)\right) \quad then \quad x^b_{id}(t+1) = xx'_{id} \\ &else \quad x^b_{id}(t+1) = x^b_{id}(t) \quad for \quad i = 1,2,...,N \end{aligned} \quad . \tag{25}$$

$\vec{\tilde{X}}$ in the binary search space is calculated as Eq. (26) and has a binary value.

$$\vec{\tilde{X}} = \vec{X}^b_j + \vec{X}^b_k + \vec{P}_{gbest} \tag{26}$$

If the position of $\vec{\tilde{X}}$ is better than the $\vec{P}_{gbest}$, the $\vec{P}_{gbest}$ is replaced by $\vec{\tilde{X}}$ in the swarm and the next position is computed.

4 Experimental results and discussion

To evaluate the performance of proposed methods, twenty minimization benchmark functions are selected [39, 40] as elucidated in Section 4.1. The results for the real and binary search spaces are presented in sections 4.2 and 4.3, respectively.

4.1 Benchmark functions

In this experimental study, twenty minimization functions are employed as detailed in Table 1. In the table, *Range* is the feasible bound and the dimension of function. F_{opt} is the optimum value of each function. The ring topology is used as the neighborhood structure in the local topology and the number of neighbors is 2. In rotated functions, the rotation increases the function complexity and does not affect the shape of function. The variable $\vec{Y}$ is computed using an orthogonal matrix M [41] and applied to obtain the fitness value of rotated function. In shifted functions, the global optimum $\vec{X}^* = (x_1^*, x_2^*, ..., x_n^*)$ is shifted to the new position $\vec{O} = (o_1, o_2, ..., o_n)$. In the hybrid function, some different basic functions are used to construct these functions. The composition function merges the properties of some functions and maintains continuity around the global/local optima [40].

Table 1: Dimensions, ranges, and global optimum values of test functions

Test function	$[Range]^n$	F_{opt}
$F_1(x)$	$[-100,100]^n$	0
$F_2(x)$	$[-10,10]^n$	0

$F_3(x)$	$[-100,100]^n$	0
$F_4(x)$	$[-100,100]^n$	0
$F_5(x)$	$[-100,100]^n$	0
$F_6(x)$	$[-1.28,1.28]^n$	0
$F_7(x)$	$[-5.12,5.12]^n$	0
$F_8(x)$	$[-32,32]^n$	0
$F_9(x)$	$[-600,600]^n$	0
$F_{10}(x)$	$[-50,50]^n$	0
$F_{11}(x)$	$[-100,100]^n$	300
$F_{12}(x)$	$[-100,100]^n$	500
$F_{13}(x)$	$[-100,100]^n$	600
$F_{14}(x)$	$[-100,100]^n$	800
$F_{15}(x)$	$[-100,100]^n$	1600
$F_{16}(x)$	$[-100,100]^n$	1700
$F_{17}(x)$	$[-100,100]^n$	2000
$F_{18}(x)$	$[-100,100]^n$	2200
$F_{19}(x)$	$[-100,100]^n$	2400
$F_{20}(x)$	$[-100,100]^n$	2500

The benchmark functions are as follows:

1. Sphere Model (Unimodal Function)

$$F_1(x) = \sum_{i=1}^{n} x_i^2$$

2. Schewefel' s Problem 2.22 (Unimodal Function)

$$F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

3. Schwefel's Problem 1.2 (Unimodal Function)

$$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$$

4. Schwefel's Problem 2.21 (Unimodal Function)

$$F_4(x) = \max_i \{|x_i|, 1 \le i \le n\}$$

5. Step Function (Unimodal Function)

$$F_5(x)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$$

6. Quartic Function i.e. Noise (Unimodal Function)

$$F_6(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$$

7. Rastrigin's Function (Multimodal Function)

$$F_7(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$$

8. Ackley's Function (Multimodal Function)

$$F_8(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)+20+e$$

9. Generalized Griewank Function (Multimodal Function)

$$F_9(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$$

10. Generalized Penalized Function (Multimodal Function)

$$F_{10}(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2]\right\}+\sum_{i=1}^{n}u(x_i,10,100,4)$$

$$y_i=1+\frac{x_i+1}{4}\quad,\quad u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a\le x_i\le a\\ k(-x_i-a)^m & x_i<-a\end{cases}$$

11. Rotated Discus Function (Multimodal Function)

$$F_{11}(x)=10^6y_i^2+\sum_{i=1}^{n}y_i^2+F_{11}^*,\quad y=M(x-o),\quad F_{11}^*=300$$

12. Shifted and Rotated Ackley's Function (Multimodal Function)

$$F_{12}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi z_i\right) + 20 + e + F_{12}^{*}, \quad z = (x-o)\times M, \quad F_{12}^{*} = 500$$

13. Shifted and Rotated Weierstrass Function (Multimodal Function)

$$F_{13}(x) = \sum_{i=1}^{n}\left(\sum_{k=0}^{k\max}\left[a^k \cos\left(2\pi b^k (z_i + 0.5)\right)\right]\right) - n\sum_{k=0}^{k\max}\left[a^k \cos\left(2\pi b^k \times 0.5\right)\right] + F_{13}^{*}, \quad z = \frac{0.5(x-o)}{100}\times M$$

$F_{13}^{*} = 600 \quad a = 0.5, b = 3, k\max = 20.$

14. Shifted Rastrigin's Function (Multimodal Function)

$$F_{14}(x) = \sum_{i=1}^{n}\left[z_i^2 - 10\cos(2\pi z_i) + 10\right] + F_{14}^{*}, \quad z = \frac{5.12(x-o)}{100}, \quad F_{14}^{*} = 800$$

15. Shifted and Rotated Expanded Scaffer's F6 Function

$$F_{15}(z) = f_{15}(z) + F_{15}^{*}, \quad F_{15}^{*} = 1600, \quad z = M(x-o) + 1$$

Scaffer's F6 Function: $g(x,y) = 0.5 + \dfrac{(\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2}$

$$f_{15}(x) = g(x_1, x_2) + g(x_2, x_3) + \ldots + g(x_{n-1}, x_n) + g(x_n, x_1)$$

16. Hybrid Function 1 (N=3) (As detailed in [40])

17. Hybrid Function 4 (N=4) (As detailed in [40])

18. Hybrid Function 6 (N=5) (As detailed in [40])

19. Composition Function 2 (N=3) (As detailed in [40])

20. Composition Function 3 (N=3) (As detailed in [40])

4.2 Comparative results in the real search space

The ICAPSO, ILCAPSO, CAPSO and LCAPSO are independently run 30 times on the functions and the results are provided in Tables 2 and 3. In these tables the average best solution, the standard deviation (SD) and the median of the best

solution in the last iteration are reported along the dimension (*n*) 30 and 50. The best results among the algorithms are shown in bold numbers. The population size is considered as 50 (N=50). The maximum iteration is 1500 for n=30, and 2000 for n=50, respectively.

Also, the Wilcoxon's rank sum test [42] is conducted in order to determine whether the results obtained by the ICAPSO/ILCAPSO algorithms are statistically different from those of CAPSO/LCAPSO with a confidence level approaching 95%. Their test results for n=30 and n=50 are shown in the last column of Table 2 and Table 3, respectively. Where '1' indicates that the proposed algorithm significantly outperforms the compared algorithm, and '0' denotes that the results of the two considered algorithms are not significantly different from each other. For Example, '1/0' shows ICAPSO significantly outperforms the CAPSO, and ILCAPSO and LCAPSO are not significantly different from each other.

Table 2: Minimization results of the benchmark functions in the real search space (Maximum Iteration=1500 and n=30)

Functions		ICAPSO	CAPSO	ILCAPSO	LCAPSO	Wilcoxon rank sum
F1	Avg. best	1.025e-096	1.446e-059	7.782e-051	1.823e-022	1/1
	SD	3.506e-096	2.901e-059	9.430e-051	1.407e-022	
	Median best	6.512e-099	4.861e-060	3.803e-051	1.415e-022	
F2	Avg. best	8.011e-053	4.664e-031	1.657e-026	1.482e-013	1/1
	SD	1.015e-052	3.812e-031	2.169e-026	5.349e-014	
	Median best	3.968e-053	3.319e-031	9.293e-027	1.419e-013	
F3	Avg. best	1.641e-040	8.462e-025	7.421e-009	1.244e-003	1/1
	SD	4.704e-040	4.399e-024	3.892e-008	1.244e-003	
	Median best	3.592e-042	9.215e-027	8.383e-012	6.482e-004	
F4	Avg. best	3.116e-034	6.961e-017	7.290e-015	4.199e-004	1/1
	SD	3.805e-034	5.895e-017	4.628e-015	1.695e-004	
	Median best	1.219e-034	5.132e-017	5.556e-015	4.072e-004	
F5	Avg. best	0.000e+000	2.967e+000	0.000e+000	0.000e+000	1/0
	SD	0.000e+000	2.266e+000	0.000e+000	0.000e+000	
	Median best	0.000e+000	3.000e+000	0.000e+000	0.000e+000	
F6	Avg. best	7.105e-004	9.546e-003	2.790e-003	1.750e-002	1/1

	SD	4.993e-004	1.833e-003	1.427e-003	4.472e-003	
	Median best	5.421e-004	9.441e-003	2.731e-003	1.751e-002	
F7	Avg. best	2.197e+001	1.087e+002	5.182e+001	9.207e+001	1/1
	SD	1.942e+001	2.923e+001	1.624e+001	1.931e+001	
	Median best	2.079e+001	1.090e+002	5.300e+001	9.656e+001	
F8	Avg. best	4.441e-015	2.656e-001	4.441e-015	4.750e-012	1/1
	SD	0.000e+000	6.154e-001	0.000e+000	2.316e-012	
	Median best	4.441e-015	2.576e-014	4.441e-015	4.021e-012	
F9	Avg. best	2.878e-003	1.109e-002	1.267e-005	3.274e-004	1/0
	SD	1.215e-002	1.696e-002	4.626e-005	8.792e-004	
	Median best	0.000e+000	3.738e-003	0.000e+000	0.000e+000	
F10	Avg. best	1.110e-002	5.107e-001	6.012e-003	1.056e-001	1/0
	SD	2.756e-002	9.424e-001	1.262e-003	4.154e-001	
	Median best	2.732e-003	1.146e-001	5.867e-003	5.660e-003	
F11	Avg. best	14108.3738	15953.3934	26410.9957	27192.7794	1/0
	SD	4.949e+003	4.978e+003	7.775e+003	7.712e+003	
	Median best	13144.8977	15314.3908	26661.7625	28308.2855	

Table 2 (*Continued)*

Functions		ICAPSO	CAPSO	ILCAPSO	LCAPSO	Wilcoxon rank sum
F12	Avg. best	520.9998	521.0025	521.0023	521.0114	0/0
	SD	4.835e-002	4.528e-002	6.048e-002	6.101e-002	
	Median best	521.0083	521.0117	521.0033	521.0211	
F13	Avg. best	623.1718	623.5873	620.1889	623.4641	0/1
	SD	3.385e+000	3.754e+000	2.407e+000	2.048e+000	
	Median best	622.4863	623.5162	619.7469	623.9165	
F14	Avg. best	926.3337	944.1518	919.4222	914.5353	1/0
	SD	1.967e+001	3.123e+001	1.410e+001	1.265e+001	
	Median best	926.5459	939.5668	919.0151	914.7816	
F15	Avg. best	1612.1525	1612.3280	1612.0763	1612.2538	0/1
	SD	4.319e-001	3.878e-001	3.230e-001	3.089e-001	
	Median best	1612.2063	1612.3753	1612.1474	1612.2731	
F16	Avg. best	952571.1706	865547.8562	725799.5862	1184674.735	0/1
	SD	5.840e+005	5.060e+005	2.990e+005	5.526e+005	
	Median best	851590.4423	864148.6902	774651.8181	1181158.039	
F17	Avg. best	14072.7383	15684.3195	15286.5974	21442.4528	0/1
	SD	4.393e+003	6.940e+003	5.505e+003	6.629e+003	
	Median best	13586.6750	15839.8635	14365.1672	20281.5333	
F18	Avg. best	2813.6436	2883.9372	2505.1214	2609.6665	1/1
	SD	2.061e+002	2.289e+002	1.009e+002	1.266e+002	
	Median best	2804.6243	2906.3276	2506.0859	2622.8188	
F19	Avg. best	2600.0070	2630.4428	2600.1053	2609.5506	1/1
	SD	3.280e-003	1.067e+001	2.499e-002	1.157e+001	
	Median best	2600.0062	2629.4798	2600.0979	2601.6504	
F20	Avg. best	2710.2351	2715.4604	2708.7592	2714.5500	1/1
	SD	6.191e+000	4.538e+000	4.987e+000	2.166e+000	
	Median best	2712.4073	2714.6165	2710.8377	2714.8415	

Table 2 shows the results of algorithms for the functions with n=30. As seen, ICAPSO and ILCAPSO present the best results in minimizing functions 1-20 compared with CAPSO and LCAPSO except for function 14. In this function, LCAPSO has a better performance than the others algorithms. According to the Wilcoxon's rank sum test in Table 2, the results of ICAPSO are statistically significantly different from the CAPSO in functions 1-11, 14, 18-20. The results of ILCAPSO are also significantly different from the LCAPSO in functions 1-4, 6-8, 13, 15-20.

The superior convergence rate of ICAPSO and ILCAPSO are illustrated in Fig. 3. The results in this Fig. show that ICAPSO and ILCAPSO tend to find the global optimum for F_8 and F_{19} faster than CAPSO and LCAPSO.

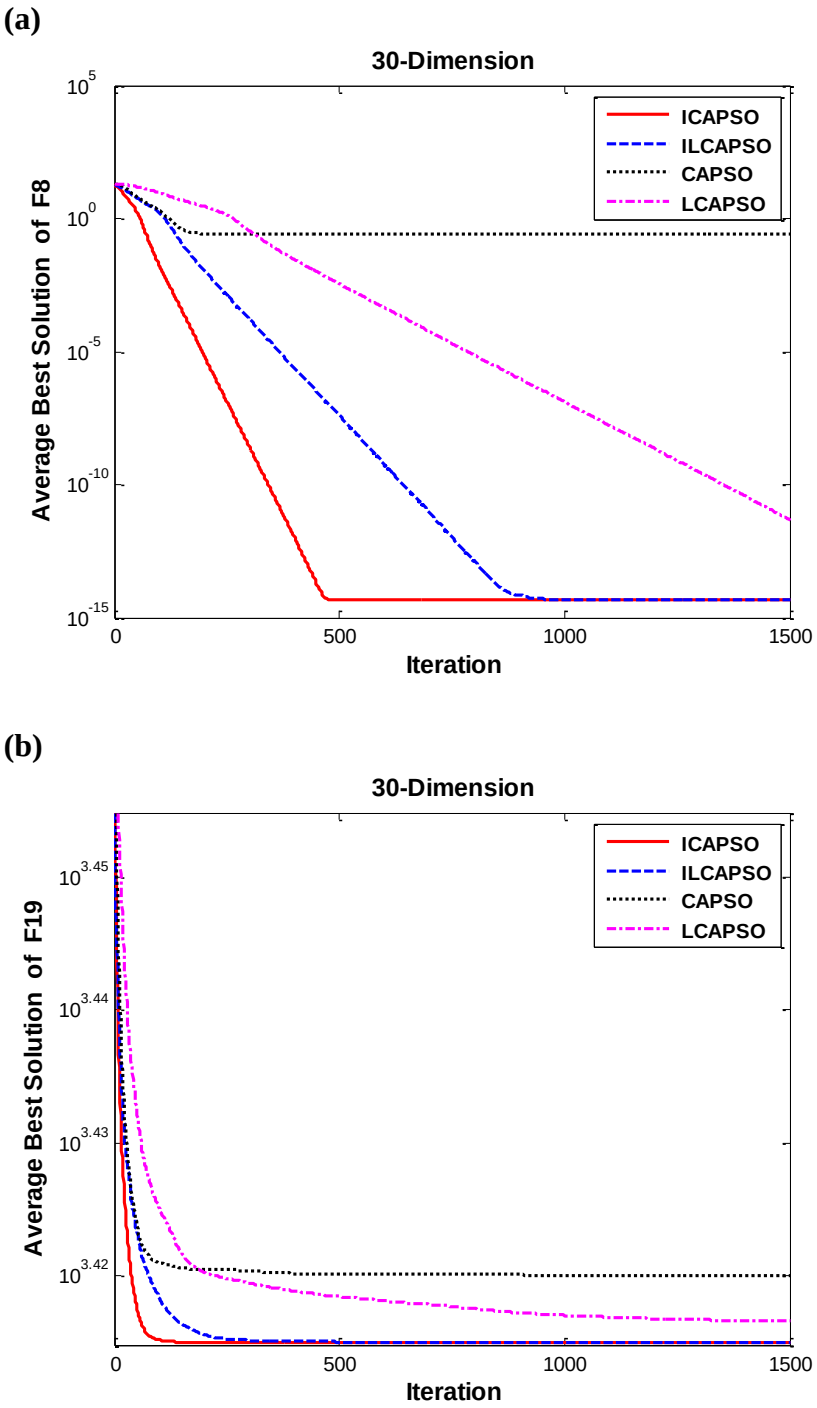


Fig. 3: Convergence performance of ICAPSO, ILCAPSO, CAPSO, and LCAPSO for the test functions (a) F_8, and (b) F_{19} (n=30).

Table 3 presents the results of the algorithms on the benchmark functions for n=50. As shown in this table, ICAPSO and ILCAPSO achieve the best results in

all of the functions, except in F_{17}. In this function, CAPSO provides the best performance. From the results in this table, it could be concluded that ICAPSO and ILCAPSO are more powerful and robust than CAPSO and LCAPSO to solve functions with high dimensions. Also, the results of Wilcoxon's rank sum test show that ICAPSO is statistically significantly different from CAPSO in functions 1-10, 12, 14, 15, 18-20. According to the tests, the results of the ILCAPSO algorithm were statistically significantly different from the LCAPSO in functions 1-4, 6-8, 10, 11, 13, 15, 17, 19, 20. Fig. 4 demonstrates the convergence performance of algorithms for F_9 and F_{20} with n=50, respectively. The proposed algorithms show considerably better results than CAPSO and LCAPSO, and ICAPSO achieves the global optimum in F_9.

Table 3: Minimization results of the benchmark functions in the real search space (Maximum Iteration=2000 and n=50).

Functions		ICAPSO	CAPSO	ILCAPSO	LCAPSO	Wilcoxon rank sum
F1	Avg. best	**8.828e-117**	1.554e-055	1.745e-054	5.829e-020	1/1
	SD	3.182e-116	2.430e-055	2.146e-054	3.198e-020	
	Median best	1.959e-118	5.409e-056	1.001e-054	5.204e-020	
F2	Avg. best	**6.168e-065**	1.178e-029	5.842e-027	6.122e-012	1/1
	SD	1.014e-064	7.347e-030	9.822e-027	2.622e-012	
	Median best	2.800e-065	1.024e-029	2.640e-027	5.266e-012	
F3	Avg. best	**4.756e-035**	7.233e-025	8.586e-005	6.488e-002	1/1
	SD	2.558e-034	1.347e-024	3.878e-004	4.861e-002	
	Median best	1.491e-039	1.029e-025	1.085e-007	5.327e-002	
F4	Avg. best	**1.438e-035**	2.920e-016	5.187e-016	4.055e-004	1/1
	SD	6.425e-035	2.521e-016	7.181e-016	1.839e-004	
	Median best	5.894e-037	1.840e-016	2.651e-016	3.546e-004	
F5	Avg. best	**0.000e+000**	6.900e+000	**0.000e+000**	**0.000e+000**	1/0
	SD	0.000e+000	3.661e+000	0.000e+000	0.000e+000	
	Median best	0.000e+000	7.000e+000	0.000e+000	0.000e+000	
F6	Avg. best	**5.362e-004**	1.029e-002	3.896e-003	2.407e-002	1/1
	SD	3.162e-004	1.480e-003	1.510e-003	4.982e-003	
	Median best	4.490e-004	1.019e-002	3.974e-003	2.357e-002	
F7	Avg. best	**1.183e+001**	2.784e+002	1.520e+002	2.265e+002	1/1
	SD	3.899e+001	4.589e+001	4.267e+001	2.823e+001	
	Median best	0.000e+000	2.703e+002	1.598e+002	2.241e+002	
F8	Avg. best	**4.441e-015**	3.642e-014	**4.441e-015**	6.472e-011	1/1
	SD	0.000e+000	6.046e-015	0.000e+000	2.563e-011	
	Median best	4.441e-015	3.997e-014	4.441e-015	6.246e-011	
F9	Avg. best	**0.000e+000**	4.884e-003	1.037e-006	1.278e-004	1/0
	SD	0.000e+000	7.171e-003	5.679e-006	5.000e-004	
	Median best	0.000e+000	1.110e-016	0.000e+000	0.000e+000	

Table 3 (*Continued)*

Functions		ICAPSO	CAPSO	ILCAPSO	LCAPSO	Wilcoxon rank sum
F10	Avg. best	3.229e-002	4.801e-001	**1.970e-002**	1.065e+000	1/1
	SD	1.866e-002	8.045e-001	1.278e-002	1.322e+000	
	Median best	2.738e-002	1.009e-001	1.705e-002	8.932e-002	
F11	Avg. best	**44485.4481**	45556.9610	70079.1786	84916.1127	0/1
	SD	9.859e+003	9.776e+003	1.076e+004	1.397e+004	
	Median best	42251.1311	45194.9125	70251.6707	87664.1160	
F12	Avg. best	**521.1699**	521.1943	521.1790	521.1724	1/0
	SD	3.322e-002	4.192e-002	5.097e-002	4.066e-002	
	Median best	521.1652	521.2025	521.1880	521.1747	
F13	Avg. best	648.4922	650.6512	**643.3649**	648.2948	0/1
	SD	4.172e+000	4.163e+000	3.711e+000	2.874e+000	
	Median best	648.6489	650.7471	642.5474	647.9378	
F14	Avg. best	1135.7346	1161.5958	**1093.8766**	1103.2892	1/0
	SD	4.275e+001	4.977e+001	2.459e+001	2.333e+001	
	Median best	1132.2285	1166.8831	1097.1013	1098.6099	
F15	Avg. best	1621.5651	1621.8709	**1621.5517**	1621.9478	1/1
	SD	4.139e-001	4.515e-001	4.134e-001	2.732e-001	
	Median best	1621.5350	1621.8913	1621.6101	1622.0002	
F16	Avg. best	**3685973.98**	3726028.2587	4227044.3724	4817027.0262	0/0
	SD	1.936e+006	2.071e+006	1.943e+006	1.696e+006	
	Median best	3597414.71	3075460.1897	4140021.0906	4823741.0294	
F17	Avg. best	19828.8129	**18139.9241**	21701.1600	30361.9161	0/1
	SD	7.156e+003	6.581e+003	6.932e+003	9.787e+003	
	Median best	20258.9878	17741.2682	20120.5632	29231.8803	
F18	Avg. best	3622.3171	3773.0502	**3323.8594**	3327.9987	1/0
	SD	3.218e+002	3.495e+002	2.346e+002	1.969e+002	
	Median best	3621.3543	3812.9733	3329.5278	3326.2228	
F19	Avg. best	**2602.8718**	2681.8543	2616.9266	2667.5182	1/1
	SD	1.540e+001	1.714e+001	2.516e+001	4.074e+000	
	Median best	2600.0061	2684.3477	2600.2308	2668.7259	
F20	Avg. best	**2717.5469**	2736.4307	2722.4879	2733.7546	1/1
	SD	1.822e+001	5.367e+000	1.290e+001	5.432e+000	
	Median best	2713.3904	2736.3127	2727.6381	2734.2494	

4.3 Comparative results in the binary search space

The IBCAPSO, ILBCAPSO, BCAPSO, and LBCAPSO algorithms are applied to the benchmark functions. In all cases, population size is set to 50 (N=50). The maximum iteration is 500 for all functions. Dimension is 10 (n=10) and each

continuous variable represents with 25 bits. Therefore, for each continuous function the dimension of particles is *dim=10*25*.

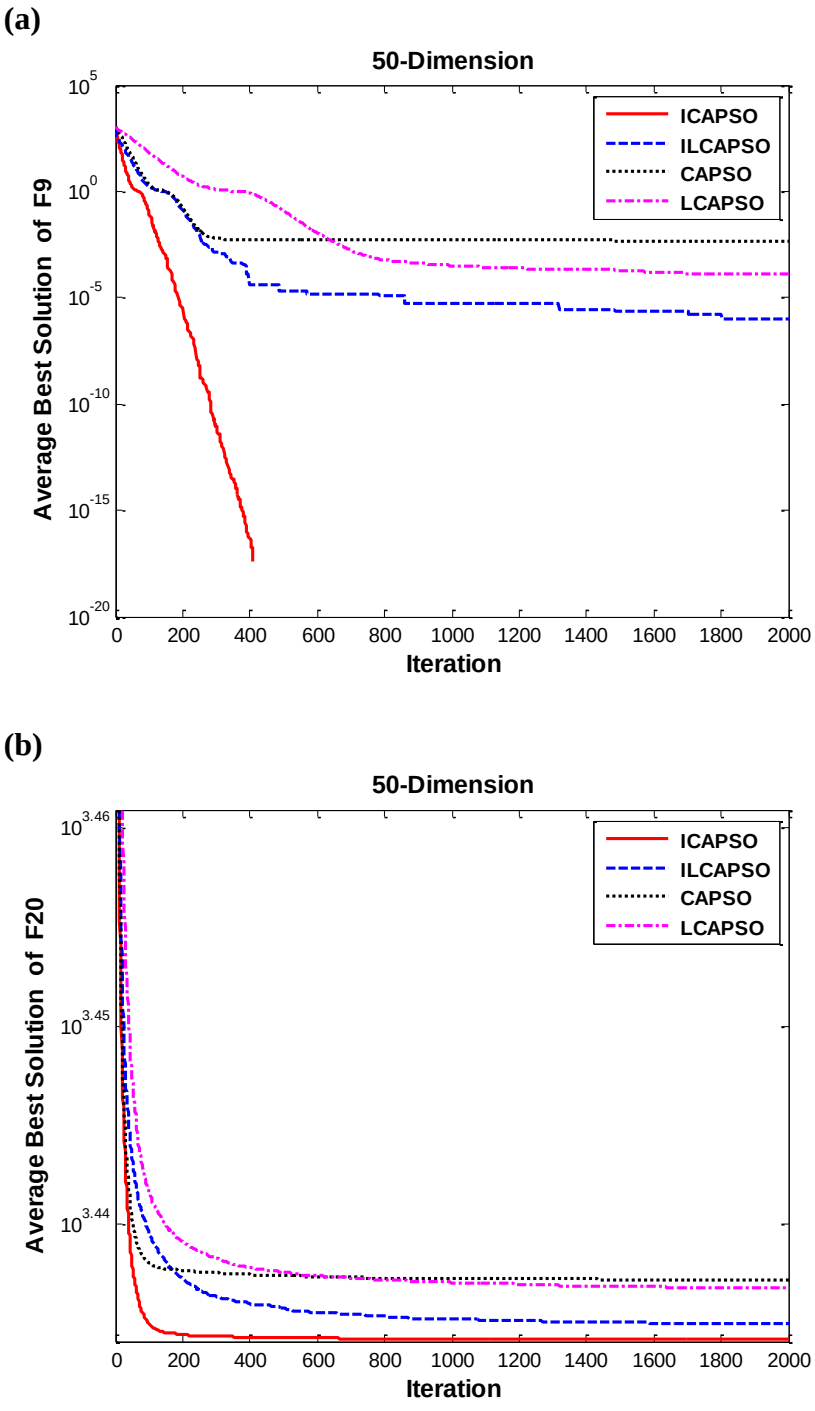


Fig. 4: Convergence performance of ICAPSO, ILCAPSO, CAPSO, and LCAPSO for the test functions (a) F_9, and (b) F_{20} (n=50)

The results are averaged over 30 independent runs under 30 different random seeds and the average best solution, SD of best solution and median of the best solution in the last iteration are reported in Table 4.

As observed, IBCAPSO reaches a much better solution than the other algorithms for unimodal functions F_1-F_6, multimodal functions F_7, F_8, F_{12}, and Hybrid Function1 (F_{16}), while ILBCAPSO presents the best solution for the other functions. The progress of the average best solution over 30 independent runs for functions F_9 and F_{20} are shown in Fig. 5. As shown, the proposed methods in these functions provide the best results.

Table 4: Minimization results of the benchmark functions in the binary search space (Maximum Iteration=500 and dim=10*25).

	Functions	IBCAPSO	BCAPSO	ILBCAPSO	LBCAPSO
F1	Avg. best	**1.138e-007**	4.903e-007	6.448e-003	6.833e-003
	SD	4.553e-007	8.470e-007	1.037e-002	1.650e-002
	Median best	8.225e-009	5.008e-008	2.145e-003	1.749e-003
F2	Avg. best	**4.153e-008**	1.179e-006	5.765e-003	2.752e-003
	SD	1.332e-007	2.939e-006	1.045e-002	3.691e-003
	Median best	7.372e-009	5.956e-008	1.294e-003	1.079e-003
F3	Avg. best	**1.839e+002**	1.901e+002	3.274e+002	3.725e+002
	SD	3.263e+002	2.819e+002	1.678e+002	3.004e+002
	Median best	5.585e+001	8.279e+001	3.318e+002	3.072e+002
F4	Avg. best	**8.921e-006**	1.542e-005	2.837e-003	3.416e-003
	SD	8.474e-006	3.738e-005	1.768e-003	3.738e-003
	Median best	6.258e-006	5.066e-006	2.823e-003	2.213e-003
F5	Avg. best	0.000e+000	0.000e+000	0.000e+000	0.000e+000
	SD	0.000e+000	0.000e+000	0.000e+000	0.000e+000
	Median best	0.000e+000	0.000e+000	0.000e+000	0.000e+000
F6	Avg. best	**4.736e-003**	6.000e-003	1.561e-002	1.672e-002
	SD	3.174e-003	5.064e-003	8.580e-003	1.020e-002
	Median best	3.863e-003	5.200e-003	1.305e-002	1.360e-002
F7	Avg. best	**6.817e+000**	6.926e+000	9.059e+000	6.829e+000
	SD	3.921e+000	2.926e+000	2.544e+000	2.187e+000
	Median best	6.159e+000	6.936e+000	8.911e+000	7.093e+000
F8	Avg. best	**3.393e-003**	5.381e-001	3.131e-001	6.836e-001
	SD	1.817e-002	9.448e-001	5.174e-001	7.397e-001
	Median best	5.147e-005	2.052e-004	9.752e-002	3.866e-001
F9	Avg. best	1.126e-001	1.279e-001	**1.110e-001**	1.392e-001
	SD	6.830e-002	7.804e-002	7.757e-002	7.872e-002
	Median best	1.033e-001	1.105e-001	9.196e-002	1.211e-001

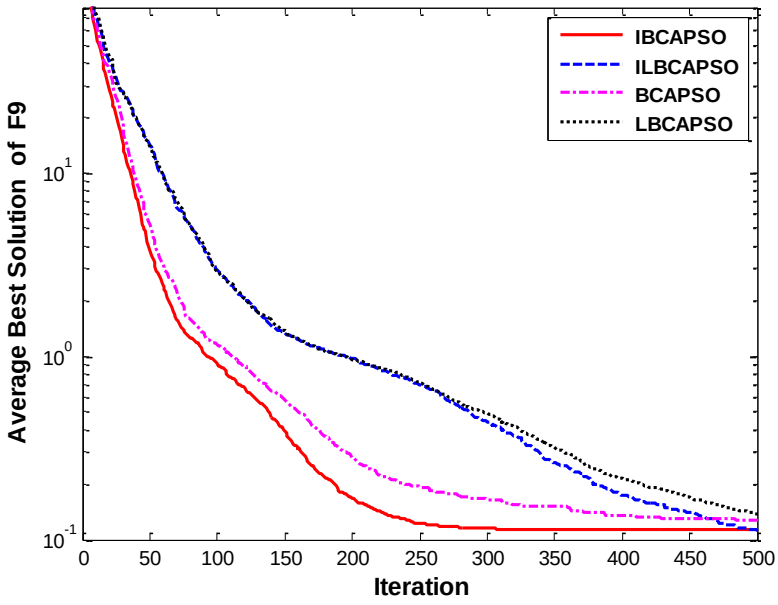
Table 4 (*Continued)*

	Functions	IBCAPSO	BCAPSO	ILBCAPSO	LBCAPSO
F10	Avg. best	7.061e-001	9.461e-001	**3.716e-001**	4.305e-001
	SD	5.796e-001	6.719e-001	3.168e-001	2.158e-001
	Median best	5.647e-001	6.426e-001	3.043e-001	4.279e-001
F11	Avg. best	4089.0325	5856.3921	**2005.8525**	2049.9160
	SD	4.195e+003	4.092e+003	9.150e+002	1.028e+003
	Median best	3019.2807	4626.5588	1634.1162	1936.4875
F12	Avg. best	**519.7544**	520.02578	519.9838	520.0219
	SD	1.507e+000	2.794e-002	1.609e-001	1.260e-002
	Median best	520.0229	520.0173	520.0203	520.0214
F13	Avg. best	603.37867	603.9220	**603.0888**	603.3670
	SD	1.139e+000	1.152e+000	7.812e-001	7.996e-001
	Median best	603.39722	603.7366	603.0529	603.3208
F14	Avg. best	809.3942	809.9129	**809.2308**	810.2645
	SD	3.755e+000	4.204e+000	2.448e+000	2.816e+000
	Median best	809.4867	809.3702	809.3076	810.4597
F15	Avg. best	1602.6232	1602.6324	**1602.4854**	1602.6187
	SD	4.413e-001	3.931e-001	3.380e-001	2.952e-001
	Median best	1602.7042	1602.6297	1602.5307	1602.6520
F16	Avg. best	**182271.7819**	247339.8280	35388.6363	39425.6865
	SD	2.895e+005	4.283e+005	2.528e+004	2.788e+004
	Median best	72846.8776	44469.9284	30273.5375	32404.8293
F17	Avg. best	6903.6596	7018.8487	**2142.9111**	2240.0725
	SD	7.121e+003	5.376e+003	1.986e+002	3.635e+002
	Median best	3358.9914	4525.4251	2073.3982	2135.3102
F18	Avg. best	2227.8825	2254.8728	**2217.1391**	2220.8319
	SD	3.083e+001	7.165e+001	6.811e+000	5.083e+000
	Median best	2221.3894	2221.8329	2220.7322	2222.0297
F19	Avg. best	2528.5092	2529.2982	**2523.3569**	2525.0597
	SD	8.865e+000	8.978e+000	4.083e+000	4.099e+000
	Median best	2526.8022	2527.7557	2522.8331	2525.1876
F20	Avg. best	2688.5455	2700.1217	**2675.9495**	2694.8788
	SD	2.617e+001	1.264e+001	2.183e+001	1.499e+001
	Median best	2701.5283	2702.3941	2679.4565	2701.6097

5 Conclusion

In this study, an improved CAPSO scheme has been introduced for global and local topologies in the real and binary search spaces. The proposed methods, ICAPSO, ILCAPSO, IBCAPSO, and ILBCAPSO are global search algorithms with several advantages which make the algorithms convenient to use.

(a)



(b)

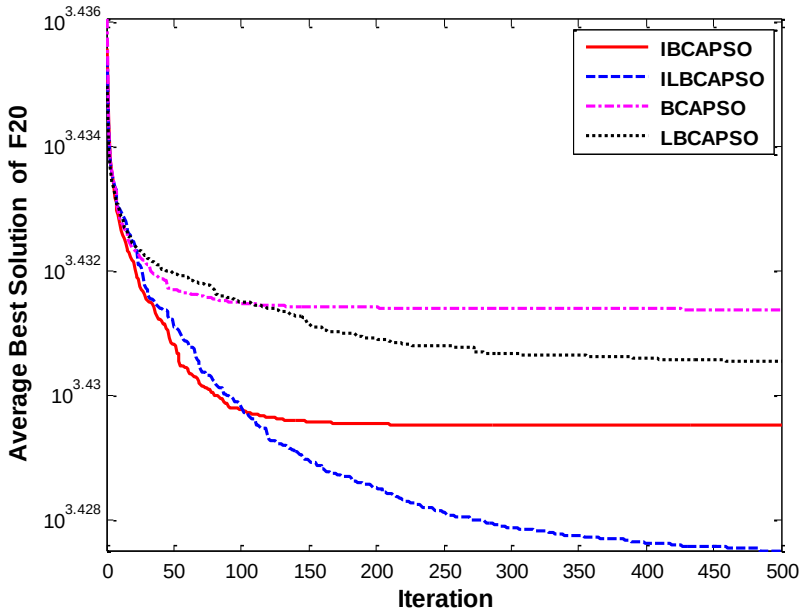


Fig. 5: Convergence performance of IBCAPSO, ILBCAPSO, BCAPSO, and LBCAPSO for the test functions (a) F_9, and (b) F_{20} (n=50)

These advantages are the ease of implementation, simple concept, and insensitive to the variable dimensions. To evaluate the performance of algorithms, a set of standard benchmarks have been employed including unimodal, multimodal,

rotated, shifted, hybrid, and composition functions. The experimental results show that the proposed methods have the better performance than CAPSO, LCAPSO, BCAPSO, and LBCAPSO, in terms of the quality of the final solutions, and the convergence rate with increasing the dimension.

ACKNOWLEDGEMENTS

The authors thank the Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for supporting in R & D, and *Soft Computing Research Group (SCRG),* Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia for the inspiration and moral support in conducting this research. Also, they hereby would like to appreciate the post-doctoral program, Universiti Teknologi Malaysia (UTM), for the financial support and research activities. This work was partially supported by the Ministry of Higher Education (MOHE) under the Fundamental Research Grant Scheme (FRGS 4F802 and FRGS 4F786).

References

[1] Kennedy J., & Eberhart R. (1995). Particle swarm optimization. *In: Proceedings of IEEE International Conference on Neural Networks,* (Vol. 4, pp. 1942–1948). IEEE.

[2] Shi Y., & Eberhart R. (1998). A modified particle swarm optimizer. *In: Proceedings of IEEE International Conference on Evolutionary Computation*, (pp. 69–73). IEEE.

[3] Gao H., Kwong S., Yang J., & Cao J. (2013). Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation. *Information Sciences, 250*, 82-112.

[4] Chen C.-H., & Liao Y.-Y. (2014). Tribal particle swarm optimization for neurofuzzy inference systems and its prediction applications. *Communications in Nonlinear Science and Numerical Simulation, 19 (4)*, 914-929.

[5] Bakshi S., Jagadev A. K., Dehuri S., & Wang G.-N. (2014). Enhancing scalability and accuracy of recommendation systems using unsupervised learning and particle swarm optimization. *Applied Soft Computing, 15*, 21-29.

[6] Ting C.-J., Wu K.-C., & Chou H. (2014). Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications, 41 (4)*, 1543-1550.

[7] Shokrian M., & High K. A. (2014). Application of a multi objective multi-leader particle swarm optimization algorithm on NLP and MINLP problems. *Computers & Chemical Engineering, 60*, 57-75.

[8] Zhan Z.-H., Zhang J., Li Y., & Chung H. H. (2009). Adaptive particle swarm optimization. *IEEE Transactions* on *Systems, Man*, and Cybernetics, *Part B, 39 (6)*, 1362-1381.

[9] Beheshti Z., & Shamsuddin S. M. (2013). A review of population-based meta-heuristic algorithms. *International Journal of Advances* in *Soft Computing* and its *Applications*. *5 (1)*, 1-35.

[10]Beheshti Z., Shamsuddin S. M., & Hasan S. (2013). MPSO: Median-oriented particle swarm optimization. *Applied Mathematics and Computation, 219 (11)*, 5817–5836.

[11]Beheshti Z., & Shamsuddin S. M. (2014). CAPSO: centripetal accelerated particle swarm optimization. *Information Sciences, 258*, 54–79.

[12]Jiang Y., Hu T., Huang C. C., & Wu X. (2007). An improved particle swarm optimization algorithm. *Applied Mathematics and Computation*, *193 (1)*, 231–239.

[13]Ratnaweera A., Halgamuge S., & Watson H. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions* on *Evolutionary Computation, 8 (3)*, 240–255.

[14]Kennedy J., & Mendes R (2002). Population structure and particle swarm performance. *in: Proceedings of IEEE international conference on Evolutionary Computation*, (Vol. 2, pp. 1671-1676). IEEE.

[15]Kennedy J., & Mendes R (2006). Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions* on *Systems, Man*, and Cybernetics, *Part C, 36 (4)*, 515-519.

[16]Parsopoulos K. E., & Vrahatis M. N. (2004). UPSO: a unified particle swarm scheme. *in: Lecture series on Computer and Computational Sciences*, (Vol. 1, pp. 868–873).

[17]Beheshti Z., Shamsuddin S. M., Sulaiman S. (2014). Fusion Global-Local-Topology Particle Swarm Optimization for global optimization problems. *Mathematical Problems* in *Engineering, 2014*, 1-19.

[18]Wang H., Wang W., Wu Z. (2013). Particle swarm optimization with adaptive mutation for multimodal optimization. *Applied Mathematics and Computation*, *221*, 296-305.

[19]Chen Y. P., Peng W.C., Jian M. C. (2007). Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Transactions* on *Systems, Man*, and Cybernetics, *Part B, 37 (6)*, 1460–1470.

[20]Pant M., Radha T., Singh V. P. (2007). A new particle swarm optimization with quadratic interpolation. *in: Proceedings of IEEE International Conference on Computational Intelligence and Multimedia Applications*, (pp.

55–60).

[21]Shinn Y. H., Hung S. L., Weei H. L., & Shinn J. H. (2008). Orthogonal particle swarm optimization and its application to task assignment problems. *IEEE Transactions* on *Systems, Man,* and Cybernetics, *Part A, 38 (2),* 288-298.

[22]Zhan Z.-H., Zhang J., Li Y., & Shi Y.-H. (2011). Orthogonal Learning Particle Swarm Optimization. *IEEE Transactions* on *Evolutionary Computation, 15 (6),* 832-847.

[23]Alatas B., Akin E., & Bedri O. (2008). Chaos embedded particle optimization algorithms. *Chaos, Solitons* & *Fractals, 40 (4),* 1715-1734.

[24]Mendes R., Kennedy J., & Neves J. (2004). The fully informed particle swarm: simpler maybe better. *IEEE Transactions* on *Evolutionary Computation, 8 (3),* 204-210.

[25]Liang J. J., & Suganthan P. N. (2005). Dynamic multi-swarm particle swarm optimizer. *in: Proceedings of IEEE Swarm Intelligence Symposium,* (pp. 124–129). IEEE.

[26]Beheshti Z., Shamsuddin S. M. (2015). Non-parametric particle swarm optimization for global optimization. *Applied Soft Computing, 28,* 345-359.

[27]Juang Y.-T., Tung S.-L., & Chiu H.-C. (2011). Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions. *Information Sciences, 181,* 4539–4549.

[28]Liang J .J., Qin A. K., Suganthan P. N., & Baskar S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions* on *Evolutionary Computation, 10 (3),* 281-295.

[29] Kennedy J., & Eberhart R. C. (1997). A discrete binary version of the particle swarm algorithm. *in: Proceedings of IEEE international conference on computational cybernetics and simulation,* (Vol. 5, pp. 4104–4108).

[30]Pampara G., Franken N., Engelbrecht A. P. (2005). Combining particle swarm optimisation with angle modulation to solve binary problems. *in: Proceedings of IEEE Congress on Evolutionary Computation,* (pp. 89-96).

[31]Sadri J., & Suen C. Y. (2006). A genetic binary particle swarm optimization model. *in: Proceedings of 2006 IEEE Congress on Evolutionary Computation, Canada,* (pp. 656–663). IEEE.

[32]Nezamabadi-pour H., Rostami-Shahrbabaki M., & Maghfoori-Farsangi M. (2008). Binary Particle Swarm Optimization: Challenges and new Solutions. *CSI Journal* on *Computer Science* and *Engineering, 6 (1),* 21-32.

[33]Chen E., Li J., & Liu X. (2011). In search of the essential binary discrete particle swarm. *Applied Soft Computing, 11,* 3260–3269.

[34]Beheshti Z., Shamsuddin S. M., & Hasan S. (2015). Memetic binary particle swarm optimization for discrete optimization problems. *Information Sciences, 299,* 58-84.

[35]Beheshti Z. (2013). *Centripetal accelerated particle swarm optimization and its applications in machine learning,* (Doctoral dissertation, Faculty of Computing, Universiti Tekhnologi Malaysia, Johor, Malaysia).

[36]Beheshti Z., Shamsuddin S. M., Beheshti E, & Yuhaniz S. S. (2014). Enhancement of artificial neural network learning using centripetal accelerated particle swarm optimization for medical diseases diagnosis. *Soft Computing, 18 (11),* 2253-2270.

[37]Beheshti Z., Shamsuddin S. M., & Yuhaniz S. S. (2013). Binary accelerated particle swarm algorithm (BAPSA) for discrete optimization problems. *Journal of Global Optimization, 57 (2),* 549-573.

[38]Beheshti Z., Firouzi M., Shamsuddin S. M., Zibarzani M, & Yusop Z. (2015). A new rainfall forecasting model using the CAPSO algorithm and an artificial neural network. *Neural* Computing & *Applications,* 1-15, DOI: 10.1007/s00521-015-2024-7.

[39]Yao X., Liu Y., & Lin G. (1999). Evolutionary programming made faster. *IEEE Transactions* on *Evolutionary Computation, 3,* 82–102.

[40]Liang J. J., Qu B. Y., & Suganthan P. N. (2013). Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. *Technical Report 201311,* Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, (pp. 1-32).

[41]Salomon R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. *BioSystems, 39 (3),* 263–278.

[42]Wilcoxon F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1 (6),* 80–83.