Modern Applied Science; Vol. 11, No. 8; 2017
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Polar Particle Swarm Algorithm for Solving Cloud Data Migration Optimization Problem

Rizik M. H. Al-Sayyed[1], Hussam N. Fakhouri[1], Ali Rodan[1] & Colin Pattinson[2]

[1] University of Jordan, King Abdullah II School for Information Technology, Amman, Jordan

[2] Leeds Beckett University, School of Computing, Creative Technologies & Engineering, Jordan

Correspondence: Rizik M. H. Al-Sayyed, King Abdullah II School for Information Technology, University of Jordan, Amman, Jordan. Tel: 962-796-101-916. E-mail: r.alsayyed@ju.edu.jo

Received: May 2, 2017 Accepted: June 12, 2017 Online Published: July 21, 2017
doi:10.5539/mas.v11n8p98 URL: https://doi.org/10.5539/mas.v11n8p98

Abstract

Particle Swarm Optimization (PSO) has proved to be a common meta-heuristic algorithm for determining the minimum value among a set of values but it is known to suffer from the local minima problem. In this paper, we propose a novel optimization algorithm called POLARPSO that enhances the behavior of PSO and avoids the local minima problem by using a polar function to search for more points in the search space. The algorithm has been tested on 23 well-known benchmark factions and the results are verified by comparing them with state of the art algorithms: Grey Wolf Optimizer (GWO), Sine Cosine Algorithm (SCA), Multi-Verse Optimizer (MVO) as well as PSO. The paper also considers a solution to the cloud data migration problem where data migrates from highly loaded nodes to less loaded nodes in a process aims at achieving a kind of load balancing. The results prove that the proposed algorithm is applicable to solve this challenging problem in cloud environment and is able to find the best node to migrate to quickly and effectively. Our empirical results show that the proposed algorithm has enhanced the PSO behavior in reaching the best solution and outperformed the other algorithms over the tested benchmarked functions.

Keywords: optimization, cloud data migration, POLARPSO, PSO, GWO, SCA, MVO

1. Introduction

Meta-heuristic algorithms are used in solving difficult optimization problems in many fields including engineering design, economics, Internet routing and cloud computing: problems such as data migration and resource scheduling. As resources are limited and cost money, an optimal feasible solution is a crucial requirement. Real-world optimization problems vary from nonlinear to multimodal, under different complex constraints. However, finding an optimal solution or even sub-optimal solutions is challenging. Many meta-heuristic algorithms have been proposed (Yang, 2011).

Swarm-based meta-heuristic algorithms have been considered as powerful optimization techniques that are inspired by animals' social behavior. Particle swarm optimization (PSO) has been known as one of the most popular optimizers and is widely used in optimization of different applications. It provides a set of candidate solutions to the problem which are called particles of a swarm where they move through the search space and guided by the best value found by other particles and by their best performances. PSO is known to suffer from the local minima problem where they found minimum value is not the global minimum one. Our proposed algorithm, the POLARPSO proved to overcome the local minima problem as will be shown in the subsequent sections.

PSO is used to solve real life computer problems including those in a cloud environment. Cloud computing data migration is a difficult task since migration requires searching for the best node to migrate data to among a very large number of nodes where each node has a cost.

Storing data in a very large, scalable and reliable environment is a big challenge to different organizations. Data migration should be determined automatically using an elastic technique to determine the optimal node for data migration. To move data to a new cloud gracefully with minimal time, cost and disruption requires finding smart methods to transport very large volumes of data. Some cloud providers (such as Amazon Web Services) supply tools (such as rsync) that help in performing this task.

In order to reduce data migration rates, the algorithm needs to determine the hot data subject to migrate. For this, the data is divided into partitions at each node of storage; in other words, the data storage nodes are further subdivided into multiple partitions. Migration uses zoning as a basic unit to monitor nodes' load. To achieve load balancing between nodes in cloud environment and based on the normalized loading value (1/n), storage nodes are classified as a collection of either in (<1/n) or out (>1/n) immigration nodes (Yushui & Jiaheng, 2015). following this classification process, the data migration will be then applied.

In this paper, we propose an enhanced meta-heuristic algorithm that modifies the behavior of PSO algorithm by following a polar scheme in reaching the optimal solution in the search space, we call the new meta-heuristic POLARPSO. The proposed POLARPSO is then used to find the best way in cloud data migration, in order to choose the best node for data migration

Applying the Polar behavior has been widely used to solve many real-life problems and applications and this inspired us to think about applying the polar behavior during the swarm movement in PSO this method enhanced the PSO behavior in finding the global minimum when tested with the benchmark functions and in solving cloud data migration problem.

The rest of this paper is organized as follows: section 2 includes most recent related work, section 3 presents the Polar PSO (POLARPSO) Optimizer, section 4 Evaluating POLARPSO on the CEC2005 Benchmarks, section 5 presents Experiment Description, Results Collection and Analysis, section 6 shows the stability in POLARPSO algorithm, section 7 introduced Cloud Data Migration Case Study, section 8 discusses convergence analysis while the conclusion and future work are drawn in section 9.

2. Related Work

The PSO idea was first introduced by Kennedy & Eberhart (1995) then improved by them in the same year .

Selvan, S. Easter et al. (2003), proposed a modified PSO algorithm that was implemented in proportional–integral–derivative (PID) controller tuning by incorporating some special features in the conventional PSO technique; its usefulness was tested in a common control application involving PID controller tuning. The results obtained for different plant transfer functions confirmed the theoretical predictions.

Yu, S. et al. (2012), proposed a new hybrid PSO (HPSO) to solve the problem that PSO often easily falls into local optima; the results have shown that HPSO has a faster convergence rate on those simple unimodal functions and superior global search ability on those multimodal functions compared to other PSO.

Binh, (2013) proposed a new hybrid particle swarm optimization algorithm for solving Multi-Area Economic Dispatch Problem (MAEDP); the results show that their proposed approaches are stable and quite effective with MAEDP.

Khanesar et al. (2007), proposed a new interpretation for the velocity of binary PSO related to the change rate particles' bits. In addition, they were able to solve the problem of choosing a w value (the inertia term) for older versions of binary PSO.

Gimmler et al. (2006) proposed a Hybrid Particle Swarm Optimization algorithm (HPSO) that combines a Particle Swarm Optimization (PSO) algorithm with the Nelder-Mead-Simplex-method (NMS) and with Powell's Direction-Set-Method (PDS). Their experimental results indicate that PSO performance can be improved by using iterative improvement algorithms.

According to Handl & Meyer (2007), algorithms can be categorized into two groups: the first group of algorithms is inspired by nature and animals' behavior such as ants' methods in building colonies and where researchers followed the same strategy; the other group of algorithms uses techniques such as clustering-methods and data mining.

Researchers were inspired by observing wasp nests (Campos et al, 2000) and bee colonies in coming up with a gravitational search algorithm which was named the Bee Colony algorithm by Pham et al., (2005). This algorithm depends on two kinds of searching for explorative purposes: local and global, and is done through agents with two different types, as can be seen in the bee colony. Bee colonies also inspired the artificial bee colony algorithm by Karaboga in 2005, which is somehow different from the previous one; it consists of employed, onlooker and scout bees: three different kinds of agents which are trying to find food in the surrounding area using varying methods.

Mirjalili et al., (2014) proposed a new meta-heuristic called Grey Wolf Optimizer (GWO) inspired by grey wolves (Canis lupus). The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Mirjalili, (2016) proposed a novel population-based optimization algorithm called Sine Cosine

Algorithm (SCA) for solving optimization problems. The SCA created multiple initial random candidate solutions and required them to fluctuate outwards or towards the best solution using a mathematical model based on. Mirjalili et al. (2016) proposed a novel nature-inspired algorithm called Multi-Verse Optimizer (MVO), based on three concepts in cosmology: white hole, black hole, and wormhole. The mathematical models of these three concepts are developed to perform exploration, exploitation, and local search, respectively.

The main meta-heuristics algorithms are summarized in Table 1 below:

Table 1. Main meta-heuristic Algorithms

Algorithm	Inventor
Genetic Algorithm (GA)	Goldberg, 1989.
Ant Colony Optimization (ACO)	Dorigo, et al., 2006.
Particle Swarm Optimization (PSO)	Kennedy &Eberhart, 1995.
Marriage in Honey Bees Optimization Algorithm (MBO)	Abbass HA, 2001.
Artificial Fish-Swarm Algorithm (AFSA)	Li X, 2003.
Termite Algorithm	Roth M, 2005
Wasp Swarm Algorithm	Pinto PC, 2007.
Monkey Search	Mucherino A & Seref O., 2007.
Bee Collecting Pollen Algorithm (BCPA)	Lu X & Zhou Y., 2008.
Cuckoo Search (CS)	Yang X-S and Deb S., 2009.
Dolphin Partner Optimization (DPO)	Shiqin et al., 2009.
Firefly Algorithm (FA)	Yang et al., 2010.
Bird Mating Optimizer (BMO)	Askarzadeh & Rezazadeh, 2012.
Krill Herd (KH)	Gandomi & Alavi, 2012.
Fruit fly Optimization Algorithm (FOA)	Pan W-T, 2012.
Grey Wolf Optimizer	Mirjalili et al., 2014
SCA: A Sine Cosine Algorithm for solving optimization problems	Mirjalili, 2016
Multi-Verse Optimizer: a nature-inspired algorithm for global optimization	Mirjalili, et al., 2016

3. Polar PSO (POLARPSO) Optimizer

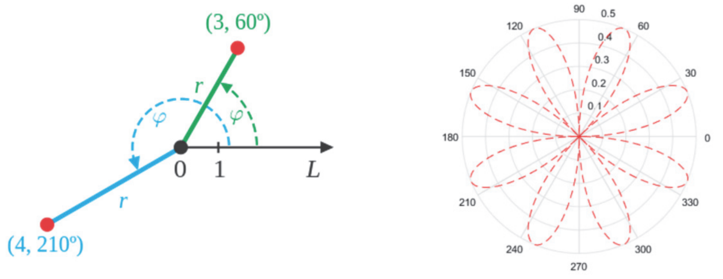
In this section we explain the inspiration of the proposed algorithm, the mathematical model and the pseudo code.

The origin of Polar Coordinates was first described by Harvard professor Julian Lowell Coolidge. Grégoire de Saint-Vincent and Bonaventura Cavalieri independently introduced the concepts in the mid-seventeenth century as described by Coolidge, (1952). Polar coordinates were first used by Cavalieri to solve a problem related to the area within an Archimedean spiral. Blaise Pascal subsequently used polar coordinates to calculate the length of parabolic arcs (Brown, et,. al., 1992). Also, Sir Isaac Newton examined the transformations between polar coordinates, which he referred to as the "Seventh Manner; For Spirals", and nine other coordinate systems (Boyer, 1949).

The system of the polar coordinate is a two-dimensional coordinate system where every point on the plane can be determined by an angle from a reference direction and a distance from a reference point. The reference point is called the pole; the polar axis is the ray from the pole in the reference direction. As shown in Figure 1 (a), the pole is 0 and the polar axis is L. We denote to the radial coordinate as an r or ρ and denote to the angular coordinate by θ, ϕ, or t.

The radial coordinate or radius is the distance from the pole (0), and the angle is called the angular coordinate. With reference to the figure, the green line's point (3, 60) has a radial coordinate of 3 units and angular coordinate of 60 degrees. The blue line's point (4, 210) has a radial coordinate of 4 units and angular coordinate of 210 degrees.

In a polar coordinate grid, see Figure 1 (b), a series of circles extend out from the pole (or origin in a rectangular coordinate grid) with five different lines passing through the pole to represent the angles at which the exact values are known for the trigonometric functions.



(a) (b)

Figure 1. (a) Polar Coordinate Representation (b) Polar Coordinate Grid (Rose Curve)

Polar coordinates have been widely used to solve many real-life problems, with applications such as collision avoidance between ships and other obstructions, calculating groundwater flow in radial symmetric wells, guiding industrial robots in various production applications, audio pickup patterns for cardioid microphones and in calculations involving aircraft navigation. All these applications for polar coordinates inspire us to think about employing this method to enhance the PSO behavior in finding global minimum and applying it to cloud data migration.

The behaviors of the polar function; as represented by equation (1); forms a rose graph curve; see Figure 1 (b).

$$\theta = 0{:}0.01{:}2*pi;$$

$$r = a*\sin(n*\theta) * \cos(n*\theta), \tag{1}$$

$$\text{where } a \neq 0 \text{ and } n \text{ is an integer} > 1$$

The formed graph is called a rose curve because the loops that are formed resemble petals. The number of petals that are presented depends on the value of ***n***. The value of ***a*** determines the length of the petal; see Figure 1 (b).

As shown in Figure 2, the PSO moves in one direction toward the global best solution; X_i^k to X_i^{k+1} and it doesn't check for the other points (V_i^k, $P_{best\,i}$, and $G_{best\,i}$) while moving.

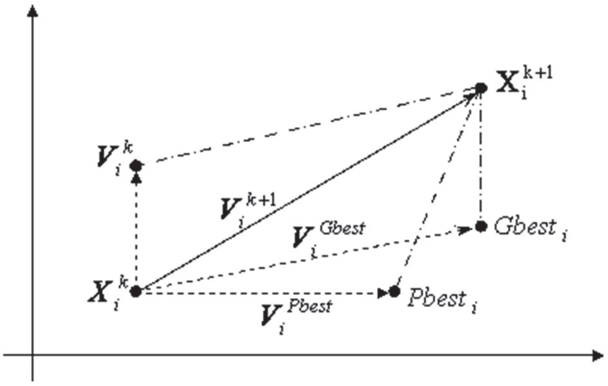


Figure 2. PSO behavior

To improve the PSO search ability we improved the search ability of the PSO at each iteration by employing the polar ability and we added it to PSO ability as described in equation (2) written in Matlab, this hybrid combination has extended the search ability in PSO to include more points and polar direction to the particle movement which increased the possibility to find the global minima and avoid the local minima problem, rather than getting stuck at one local minima point the polar behavior bypass this obstacle.

$$\text{New participle position}=\sin(2*pBest*pi*.01).*\cos(2*pBest*pi*.01)*rand()*.01 \tag{2}$$

Where pBest is the local best value found at each iteration of PSO

$$Pi = 3.1415$$

rand(): is a random function that generate numbers between 0 and 1

POLARPSO increases the ability of exploitation and exploration by providing a co-rotating frame during a particle's motion. To define a co-rotating frame, an origin is selected first from the distance r(t) to the particle ahead. An axis of rotation is set up that is perpendicular to the plane of motion of the particle that passes through the origin. Then, at the selected moment t; the PSO particle is moved to new position in a rotation movement that explores many points P1 through P8 ahead and explores more areas; see Figure 3.

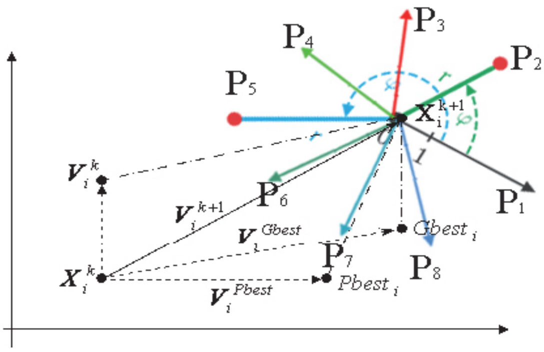


Figure 3. POLARPSO behavior at each iteration

The pseudo code for POLARPSO is shown in Figure 3. This algorithm enhanced the original standard PSO algorithm that was provided by Clerc, M. (2012) as shown in Figure 3 by adding the lines 17 to 21.

```
1- for each particle i = 1, ..., S do
2- Initialize the particle's position with a uniformly distributed random vector: xi ~ U(blo, bup)
3- Initialize the particle's best known position to its initial position: pi ← xi
4- if f(pi) < f(g) then
5- update the swarm's best known position: g ← pi
6- Initialize the particle's velocity: vi ~ U(-|bup-blo|, |bup-blo|)
7- while a termination criterion is not met do:
8- for each particle i = 1, ..., S do
9- for each dimension d = 1, ..., n do
10- Pick random numbers: rp, rg ~ U(0,1)
11- Update the particle's velocity: vi,d ← ω vi,d + φp rp (pi,d-xi,d) + φg rg (gd-xi,d)
12- Update the particle's position: xi ← xi + vi
13- if f(xi) < f(pi) then
14- Update the particle's best known position: pi ← xi
15- if f(pi) < f(g) then
16- Update the swarm's best known position: g ← pi
17- Update the particle's position according to equation (2)
18- if f(xi) < f(pi) then
19- Update the particle's best known position: pi ← xi
20- if f(pi) < f(g) then
21- Update the swarm's best known position: g ← pi
```

Figure 3. Enhanced PSO Pseudo Code (POLARPSO)

4. Evaluating POLARPSO on the CEC2005 Benchmarks

This section presents an empirical evaluation of POLARPSO We evaluated the performance of POLARPSO on the CEC2005 Special Session on Real-Parameter Optimization benchmark suite and compare POLARPSO to state-of-the-art GWO, SCA, MVO and the well-known PSO optimization algorithm. The CEC2005 benchmark set consists of 23 test functions table 1. The mathematical formulations of the Functions from F1 to F5 which represent Unimodal Functions are shown in table 2. F6 to F12 represent Multimodal Functions shown in table 3. F13 to F14 are Expanded Functions shown in table 4. Finally, F15 to F23 shown in table 5 represent Hybrid Composition Functions combining multiple test problems into a complex landscape.

Table 1. Classification of CEC2005 Benchmark Functions

Unimodal Functions (1- 5)	**Unimodal Functions**	F1: Shifted Sphere Function F2: Shifted Schwefel's Problem 1.2 F3: Shifted Rotated High Conditioned Elliptic Function F4: Shifted Schwefel's Problem 1.2 with Noise in Fitness F5: Schwefel's Problem 2.6 with Global Optimum on Bounds
Multimodal Functions (6- 23)	**Basic Functions**	F6: Shifted Rosenbrock's Function F7: Shifted Rotated Griewank's Function without Bounds F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds F9: Shifted Rastrigin's Function F10: Shifted Rotated Rastrigin's Function F11: Shifted Rotated Weierstrass Function F12: Schwefel's Problem 2.13
	Expanded Functions	F13: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2) F14: Shifted Rotated Expanded Scaffer's F6
	Hybrid Composition Functions	F15: Hybrid Composition Function F16: Rotated Hybrid Composition space Function F17: Rotated Hybrid Composition Function with Noise in Fitness F18: Rotated Hybrid Composition Function F19: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum F20: Rotated Hybrid Composition Function with the Global Optimum on the Bounds F21: Rotated Hybrid Composition Function F22: Rotated Hybrid Composition Function with High Condition Number Matrix F23: Non-Continuous Rotated Hybrid Composition Function

Table 2. Unimodal Functions mathematical formulation

Function	**Formula**
1.	$f1(x) = \sum_{i=1}^{n} x_i^2$
2.	$f2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
3.	$f3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_j \right)^2$
4.	$f4(x) = \max i\{\lvert x_i \rvert, 1 \leq i \leq n\}$
5.	$f5(x) = \sum_{i=1}^{n=1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$

Table 3. Multimodal Basic Functions

6.	$f6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$
7.	$f7(x) = \sum_{i=1}^{n} ix_i^4 + \mathrm{random}(0,1)$
8.	$f8(x) = \sum_{i=1}^{n} -x_i \sin\sqrt{\lvert x_i \rvert} \; * \; \sum_{i=1} ix_i^4 * \mathrm{random}(0,1) \; *$

9. $$\mathrm{f9(x)} = \sum\nolimits_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$$

10. $$\mathrm{f10(x)} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum\nolimits_{x=i}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$$

11. $$\mathrm{f11(x)} = \frac{1}{4000}\sum\nolimits_{i=1}^{n} x_i^2 - \prod\nolimits_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

12. $$f12(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n=1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10{,}100{,}4)$$

$$y_i = 1 + \frac{x_i + 1}{4}$$

$$u(x_i, a, k, m) = \begin{Bmatrix} k(x_i - a)^m \; x_i > a \\ 0 \; -a < x_i < a \\ k(-x_i - a)^m \; x_i < a \end{Bmatrix}$$

Table 4. Multimodal expanded Functions

13. $$f13(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5{,}100{,}4)$$

14. $$f14(x) = -\sum\nolimits_{i=1}^{n}\sin(x_i).\left(\sin\left(\frac{ix_i^2}{\pi}\right)\right)^{2m}$$, m=10

Table 5 . Hybrid Composition Functions

15. $$f15(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 x_4}\right]$$

16. $$f16(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

17. $$F17(X) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos X_1 + 10$$

18. $$f18(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]\, x\, [30 + (2x_1 - 3x_2)^2 x\, (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$$

19. $$f19(x) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$$

20. $$f20(x) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$$

21. $$f21(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + C_i]^{-1}$$

22. $$f22(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + C_i]^{-1}$$

23. $$f23(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + C_i]^{-1}$$

5. Experiment Description and Results Collection and Analysis

We performed our evaluation on the 23 CEC2005 benchmark functions. The maximum number of iterations performed was 1000. The number of runs per problem was 30, and the average performance and standard deviation of these runs were collected for evaluation purposes. See Table 6 which describes the Experimental parameters of the function.

Table 6. Parameters for Experimented Functions

Function	Lower bound	Upper bound	Dimension	F Min
F1	-100	100	30	0
F2	-10	10	30	0
F3	-100	100	30	0
F4	-100	100	30	0
F5	-30	30	30	0
F6	-2400	2400	30	0
F7	-1.28	1.28	30	0
F8	-500	100	30	-418.9829x5
F9	-5.12	5.12	30	0
F10	-32	32	30	0
F11	-600	600	30	0
F12	-2400	2400	30	0
F13	-2400	2400	30	0
F14	-5	5	2	1
F15	-5	5	4	0.00030
F16	-5	5	2	1.0316
F17	[-5,0]	[10,15]	2	0.398
F18	-2	2	2	3
F19	0	1	3	-3.86
F20	-5	5	6	-3.32
F21	0	10	4	-10.1532
F22	0	10	4	-10.4028
F23	0	10	4	-10.5363

We compared the results recorded of POLARPSO with four functions; namely: PSO, GWO, SCA, and MVO, on the CEC2005 benchmark set problems, especially the new hybrid functions. The results and the overall results on all 23 functions are shown in Tables 7 to 9. Due to space, the table only shows the aggregated results of comparing each algorithm against POLARPSO.

Table 7. MEAN, STD, MIN, and MAX values over 30 Experiments for POLARPSO and PSO

	POLARPSO				**PSO**			
	MEAN	**STD**	**MIN**	**MAX**	**MEAN**	**STD**	**MIN**	**MAX**
F1	7E-122	3.8E-121	5.8E-162	2.1E-120	8.41E-18	1.08E-17	1.79E-19	3.75E-17
F2	8.88E-59	4.79E-58	1.55E-74	2.63E-57	1.42E-08	4.75E-08	8.37E-11	2.6E-07
F3	6.1E-115	3.3E-114	1.2E-145	1.8E-113	1.572928	0.856245	0.426768	3.660063
F4	3.47E-58	1.73E-57	2.71E-73	9.48E-57	0.153126	0.057302	0.063917	0.283493
F5	24.10345	0.459677	23.22311	25.26795	52.88656	27.94947	13.31383	103.6437
F6	0.008699	0.039221	1.06E-09	0.213869	1.25E-17	3.55E-17	3.01E-20	1.95E-16
F7	1.35E-05	1.02E-05	3.26E-07	3.83E-05	0.023079	0.01037	0.00663	0.046117
F8	-3216.19	506.3645	-4148.58	-2127.9	-6983.88	645.3402	-8522.6	-5521.17
F9	0	0	0	0	27.32976	6.088181	16.9143	40.79327
F10	8.88E-16	0	8.88E-16	8.88E-16	2.23E-09	2E-09	2.65E-10	6.87E-09
F11	0	0	0	0	0.010427	0.009673	0	0.032045
F12	7.35E-06	2.83E-05	1.44E-10	0.000144	9.41E-20	2.23E-19	9.16E-22	1.12E-18
F13	1.034037	0.942788	0.010989	2.963	0.000366	0.002006	2.84E-20	0.010987
F14	6.895737	4.381316	0.998009	12.67051	1.031138	0.181484	0.998004	1.992031
F15	0.000307	6.18E-19	0.000307	0.000307	0.000571	0.000309	0.000307	0.001081
F16	-1.03163	6.25E-16	-1.03163	-1.03163	-1.03163	6.78E-16	-1.03163	-1.03163
F17	0.397887	0	0.397887	0.397887	0.397887	0	0.397887	0.397887
F18	3	1.66E-15	3	3	3	2.18E-15	3	3
F19	-3.86278	2.64E-15	-3.86278	-3.86278	-3.86278	2.71E-15	-3.86278	-3.86278
F20	-3.25462	0.059923	-3.322	-3.2031	-3.26651	0.060328	-3.322	-3.2031
F21	-9.98327	0.930764	-10.1532	-5.0552	-7.61939	2.577172	-10.1532	-5.0552
F22	-10.376	0.135834	-10.4029	-9.65956	-9.52255	2.00228	-10.4029	-5.08767
F23	-10.1706	1.300869	-10.5364	-5.12848	-9.64295	2.031992	-10.5364	-5.17565

Table 8. MEAN, STD, MIN, MAX values over 30 experiments for GWO, SCA

	GWO				SCA			
	MEAN	STD	MIN	MAX	MEAN	STD	MIN	MAX
F1	7.81E-94	1.57E-93	1.64E-96	6.49E-93	4.0634E-06	8.65E-08	223.3377	1.881502
F2	1.33E-53	2.83E-53	7.84E-55	1.6E-52	8.83042E-06	2.67E-07	248.8871	1.913471
F3	1.5E-29	8.27E-29	1.89E-36	4.6E-28	3.95422E-11	1.42E-11	0.506488	0.273875
F4	2.5E-24	1.96E-24	1.1E-25	7.26E-24	3.49693E-05	1.46E-06	865.3114	8.817002
F5	25.89532	0.805491	24.28707	27.10698	4.0634E-06	8.65E-08	223.3377	1.881502
F6	0.088737	0.15214	4.15E-06	0.501891	8.83042E-06	2.67E-07	248.8871	1.913471
F7	0.000164	8.09E-05	3.52E-05	0.000366	3.95422E-11	1.42E-11	0.506488	0.273875
F8	-6646.72	665.4916	-7899	-5230.34	3.49693E-05	1.46E-06	865.3114	8.817002
F9	0	0	0	0	4.0634E-06	8.65E-08	223.3377	1.881502
F10	9.03E-15	2.09E-15	7.99E-15	1.51E-14	8.83042E-06	2.67E-07	248.8871	1.913471
F11	0.000981	0.003112	0	0.012634	3.95422E-11	1.42E-11	0.506488	0.273875
F12	0.012597	0.013294	1.96E-07	0.06579	3.49693E-05	1.46E-06	865.3114	8.817002
F13	0.112092	0.098292	3.45E-06	0.412478	4.0634E-06	8.65E-08	223.3377	1.881502

F14	1.254017	0.676136	0.998004	2.982105	8.83042E-06	2.67E-07	248.8871	1.913471
F15	0.001014	0.003598	0.000307	0.020363	3.95422E-11	1.42E-11	0.506488	0.273875
F16	-1.03163	1.27E-09	-1.03163	-1.03163	3.49693E-05	1.46E-06	865.3114	8.817002
F17	0.397887	2.2E-07	0.397887	0.397888	4.0634E-06	8.65E-08	223.3377	1.881502
F18	3	3.62E-07	3	3.000002	8.83042E-06	2.67E-07	248.8871	1.913471
F19	-3.86174	0.002643	-3.86278	-3.8549	3.95422E-11	1.42E-11	0.506488	0.273875
F20	-3.23261	0.077225	-3.32199	-3.02424	3.49693E-05	1.46E-06	865.3114	8.817002
F21	-9.82704	1.261739	-10.1531	-5.10059	4.0634E-06	8.65E-08	223.3377	1.881502
F22	-10.2326	0.947267	-10.4029	-5.1286	8.83042E-06	2.67E-07	248.8871	1.913471
F23	-10.5362	0.000107	-10.5364	-10.536	3.95422E-11	1.42E-11	0.506488	0.273875

Table 9. MEAN, STD, MIN, MAX values over 30 experiments for MVO

	MVO			
	MEAN	STD	MIN	MAX
F1	0.06987	0.01721	0.033928	0.103761
F2	0.163783	0.031259	0.105083	0.237172
F3	3.734672	1.437699	1.190944	6.932981
F4	0.299265	0.122061	0.114662	0.61259
F5	110.0559	342.5613	24.32757	1909.894
F6	0.070938	0.020428	0.031562	0.116613
F7	0.005609	0.002286	0.002963	0.012472
F8	-8140.38	840.3777	-9523.94	-5916.09
F9	97.01212	23.66112	44.79188	130.3767
F10	0.289247	0.511412	0.055902	2.131228
F11	0.249762	0.065966	0.140284	0.393325
F12	0.673869	0.819071	0.000176	2.700481
F13	0.012404	0.009557	0.003528	0.053325
F14	0.998004	1.52E-12	0.998004	0.998004
F15	0.00114	0.003642	0.000308	0.020363
F16	-1.03163	1.55E-08	-1.03163	-1.03163
F17	0.397887	2.3E-08	0.397887	0.397887
F18	3	1.84E-07	3	3.000001
F19	-3.86278	3.39E-08	-3.86278	-3.86278
F20	-3.26252	0.060492	-3.322	-3.20295
F21	-7.79302	2.808354	-10.1532	-2.63047
F22	-9.69695	1.830623	-10.4029	-5.08766
F23	-9.205	2.487936	-10.5364	-2.80663

To evaluate the effectiveness of any meta-heuristic algorithm in finding global minima, three major comparison features should be focused on: Exploitation Feature, Exploration Feature and Avoiding Local Minima. CEC2005 benchmark functions test set has functions that cover these three types, regarding the Exploitation Feature. It can be seen from the results in Table 3 that the POLARPSO algorithm has shown very competitive results in functions F1, F2, F3, F4 and F5 with all other algorithms (GWO, SCA, PSO, and MFO). Since the unimodal functions are suitable for benchmarking exploitation, therefore, the results illustrated that POLARPSO is better than the compared algorithms in terms of optimum exploitation.

Regarding the Exploration Feature, since the multimodal functions F8 to F23 have many local optima with the number increasing exponentially with dimension, then these functions are suitable to test the exploration behavior of the algorithm. The results of Table 12 to 17 of the compared algorithms showed very strong results for the POLARPSO over other algorithms for the basic and the expanded multimodal benchmark functions and excellent results for the Hybrid Composition benchmark functions. Thus it can be inferred that POLARPSO has a very good exploration feature.

Regarding Avoidance Local Minima, the Hybrid Composition benchmark functions F15 to F23 are classified as very challenging for any meta-heuristic algorithm because they have a large number of local optima values that have to be avoided. A good algorithm has to avoid them all to reach the global minima, the results showed that the POLARPSO exhibits a very competitive result in local minima avoidance, and exploitation features in the Hybrid Composition benchmark functions.

We compared the algorithms and the obtained results for the unimodal benchmark functions F1 to F7 are classified in Table 10, multimodal expanded benchmark functions F8 to F12 are classified in Table 12, on multimodal Basic benchmark functions F13 to F14 are classified in Table 14, Hybrid Composition benchmark functions F15 to F23 are classified in table 16. If POLARPSO optimal value better or equal to the compared algorithms the word "yes" is written and if the compared algorithm is better the word "No" is written

To have a fair comparison, we conducted the student t test nonparametric statistical test over these 30 runs in order to draw a statistically meaningful conclusion. This statistical test must be done due to the stochastic nature of meta-heuristics; results are in tables 11, 13, 15, and 17.

Table 10. Results of comparing PLOARPSO to PSO, GWO, SCA and MVO over the unimodal benchmark functions F1 to F5

	VS PSO	VS GWO	VS SCA	VS MFO	VS MVO
F1	Yes	Yes	Yes	Yes	Yes
F2	Yes	Yes	Yes	Yes	Yes
F3	Yes	Yes	Yes	Yes	Yes
F4	Yes	Yes	Yes	Yes	Yes
F5	Yes	Yes	Yes	Yes	Yes

Table 11. p values of the student T test over 30 runs ($p > 0.05$ have been underlined) for the unimodal benchmark functions F1 to F5

	PSO	GWO	SCA	MVO
F1	0.000195	0.009492	0.01748	9.11E-20
F2	0.111587	0.014503	0.086914	7.64E-23
F3	5.74E-11	0.321825	3.21E-05	1.3E-14
F4	6.31E-15	7.32E-08	8.69E-06	5.62E-14
F5	4.19E-06	3.1E-10	3.7E-23	0.179839

POLARPSO showed better results over PSO, GWO, SCA, and MVO in all the unimodal benchmark functions F1 to F5 as shown in Table 10. Also, the results were almost significant for all of the functions as shown in table 11.

The results of Table 10 show that the proposed algorithm is able to provide very competitive results on the unimodal test functions. The p values in Table 11 also prove that the competitive results are significant in the majority of the results. This testifies that the proposed algorithm has high exploitation ability.

Table 12. Results of comparing PLOARPSO to PSO , GWO , SCA and MVO over the multimodal Basic benchmark functions F6 to F12

	VS PSO	VS GWO	VS SCA	VS MVO
F6	No	Yes	Yes	Yes
F7	Yes	Yes	Yes	Yes
F8	No	No	No	No
F9	Yes	Yes	Yes	Yes
F10	Yes	Yes	Yes	Yes
F11	Yes	Yes	Yes	Yes
F12	No	Yes	Yes	Yes

Table 13. p values of the student T test over 30 runs (p > 0.05 have been underlined) for multimodal Basic benchmark functions F6 to F12

	PSO	GWO	SCA	MVO
F6	0.234228	0.005093	2.97E-35	2.92E-09
F7	6.33E-13	4.2E-11	4.25E-07	5.84E-14
F8	2.55E-19	8.38E-22	4.1E-11	4.43E-23
F9	5.7E-21	N/A	0.034061	6.95E-20
F10	1.19E-06	3.02E-19	6.49E-05	0.004302
F11	2.07E-06	0.089517	0.041795	6.14E-19
F12	0.164808	1.45E-05	2.5E-18	9.98E-05

POLARPSO showed better results over PSO in 4 functions, over GWO in 6 functions, over SCA in 6 functions and over MFO and MVO in 6 functions as shown in table 12. The results were all significant for student t test with SCA, MVO and only three functions showed no significant values for t test with GWO and PSO as shown in table 13

POLARPSO showed better results over SCA in 1 function.

Table 14. Results of comparing PLOARPSO to PSO, GWO, SCA and MVO over the Multimodal expanded Functions F13 to F14

	VS PSO	VS GWO	VS SCA	VS MVO
F13	NO	NO	Yes	NO
F14	NO	NO	NO	NO

Table 15. p values of the student T test over 30 runs (p > 0.05 have been underlined) the Multimodal expanded Functions F13 to F14

	PSO	GWO	SCA	MVO
F13	1.58E-06	8.78E-06	1.38E-05	1.86E-06
F14	4.37E-08	1.67E-07	4.01E-08	4.01E-08

POLARPSO didn’t show better values of the results when over function F13 and F14 as shown in table 15.

After discussing the exploitation feature of POLARPSO in the unimodal test functions, we are going to discuss the exploration feature of POLARPSO. To discuss this we will refer to the results obtained from multi-model test function Table 12 to 15. The results of this table show that the proposed algorithm is able to provide a very good exploration behavior. It may be observed that this POLARPSO is better than other algorithms on F7, F9, F10, F11 and F12. The p values suggest that the significant of the POLARPSO algorithm is results in most factions. This indicates that that POLARPSO results are competitive in exploration feature.

Table 15. Results of comparing PLOARPSO to PSO, GWO, SCA and MVO over the Hybrid Composition benchmark functions F15 to F23

	VS PSO	VS GWO	VS SCA	VS MVO
F15	Yes	Yes	Yes	Yes
F16	Yes	Yes	Yes	Yes
F17	Yes	Yes	Yes	Yes
F18	Yes	Yes	Yes	Yes
F19	Yes	Yes	Yes	Yes
F20	No	Yes	Yes	No
F21	Yes	Yes	Yes	Yes
F22	Yes	Yes	Yes	Yes
F23	Yes	No	Yes	Yes

Table 17. p values of the student T test runs ($p > 0.05$ have been underlined) for the Hybrid Composition Functions F15 to F23

	PSO	GWO	SCA	MVO
F15	6.28E-05	0.283582	0.001047	0.220563
F16	1	0.000103	5.21E-08	6.11E-05
F17	#DIV/0!	0.001814	1.22E-06	4.87E-08
F18	2.05E-06	1.24E-05	9.98E-05	2.14E-05
F19	1	0.03555	6.62E-14	8.36E-07
F20	0.414616	0.159346	8.07E-08	0.60288
F21	0.000111	0.576711	1.87E-12	0.000561
F22	0.027881	0.410475	7.18E-13	0.053326
F23	0.266077	0.134508	2.55E-10	0.086592

For the proposed algorithm; POLARPSO, outperformed PSO in 8 functions, GWO in 8 functions, better than SCA in all functions and MVO in 8 functions. The significant of the results in student T student are shown in table 17.

After discussing the Exploration feature in the Multimodal Functions we will discuss local optima avoidance. In fact, stagnation in local solutions can be resolved by promoting exploration. In addition to the discussion provided in the preceding paragraphs, local optima avoidance of POLARPSO is also competitive as seen in the results of the hybrid Composite test functions that provide a balance between exploration and exploitation features, due to the difficulty of this set of test functions. The results of POLARPSO algorithm were very competitive in this set of functions as seen in table 16.

6. Stability of the POLARPSO

The standard deviation values for all functions from F1 to F23 shown in Table 18. Except for F8 are all near the zero and this means that POLARPSO is stable for the 30 experiments that were performed, the reason for the high standard deviation value of F8 is due to the very large values in the search space; see Figure 4.

Table 18. The standard deviation vales functions F1 to F23

	std		**std**		**std**
F1	3.8E-121	F11	0	F21	0.930764
F2	4.79E-58	F12	2.83E-05	F22	0.135834
F3	3.3E-114	F13	0.942788	F23	1.300869
F4	1.73E-57	F14	4.381316		
F5	0.459677	F15	6.18E-19		
F6	0.039221	F16	6.25E-16		
F7	1.02E-05	F17	0		
F8	506.3645	F18	1.66E-15		
F9	0	F19	2.64E-15		
F10	0	F20	0.059923		

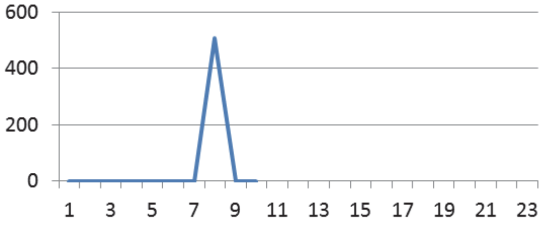


Figure 4. Plot of the Standard Deviation Vales for all Functions: F1 to F23

7. Cloud Data Migration Case Study

In cloud environments, data migrate between nodes is a process meant to achieve a kind of system load balancing. For this and based on the number of nodes compared to the overall cloud load, storage nodes are grouped as either "in_set" or "out_set". Data is moved out from heavy loaded nodes “move-out-nodes” into other less loaded nodes “move-in-nodes” to achieve an overall balanced load; this process is called Cloud Data Migration; (Yushui et al., 2016).

Finding the optimal nodes that are eligible to migrate, and their corresponding destination is not an easy task in a cloud since it is a dynamic environment that needs a clever algorithm to help, hence the POLARPSO.

To solve cloud data migration, we assumed that each data node represents a particle that needs to migrate. Each node has a position and a cost. We initialized the n data nodes (N1 to Nn) with random values and these values represent the positions of the node, we calculated the fitness value for each node based on equation (3) (Yushui & Jiaheng, 2015) using the proposed PLOARPSO algorithm (see Figure 3). Figure 5 illustrates how data migrates, Figure 5 (a) shows nodes before migration and Figure 5 (b) shows nodes during migration.

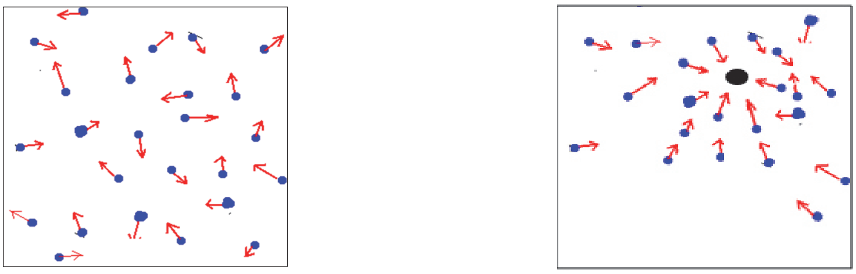


Figure 5. (a) Nodes before migration (b) Nodes during data migration

The fitness function for a migration node ***i*** is given in equation (3):

$$F_i(x_1, x_2, x_3) = x_1^2 + x_{21}^2 + x_{13}^2 \qquad (3)$$

Where x_1 represents the throughput rate of network (the amount of data transmission on the network per unit of time),

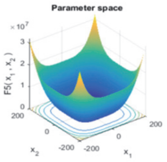
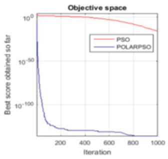
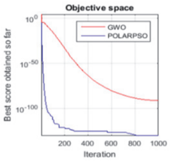
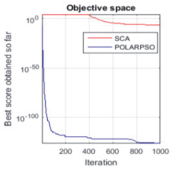
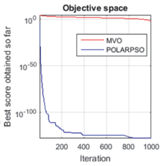
x_2 represents the accommodation in the process of the moving nodes,

x_3 represents the distance between partition of migration,

While particles move in their migration space following their target, they go toward the node with the minimal cost (the node with the smallest fitness value). The fitness value of each particle is compared with Pbest, if the current position is better; it becomes the new best position Pbest. The node with the best fitness value will become the targeted node that the data will migrate to. Particles move according to the POLARPSO algorithm and each time they go toward the minimum cost node.

We compared the results of POLARPSO in solving the cloud data migration problem with PSO, GWO, SCA, and MVO algorithms. POLARPSO showed better results over the four algorithms, the convergence graph (see Table 19) showed that POLARPSO outperformed the other four algorithms. Significantly it is the fastest to find the migration node as the speed is a very important factor in cloud data migration.

Table 19. Converge Curve for the Cloud Fitness Function of the Migrated Nodes

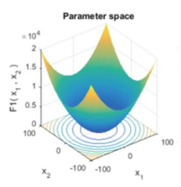
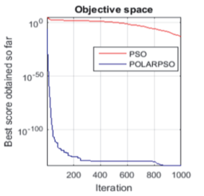
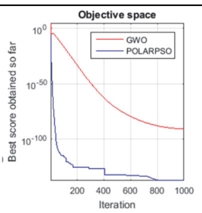
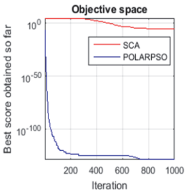
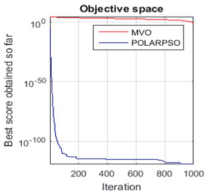
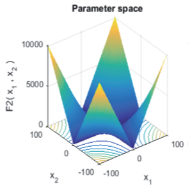
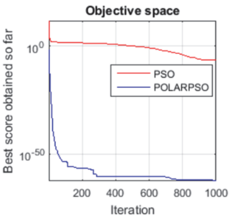
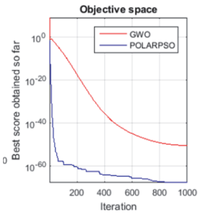
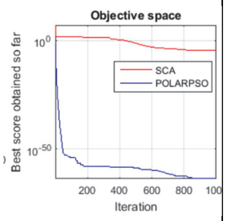
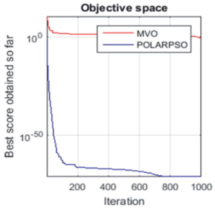
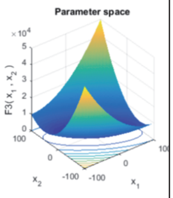
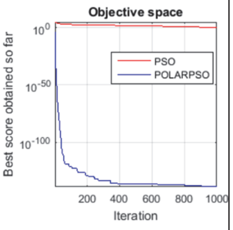
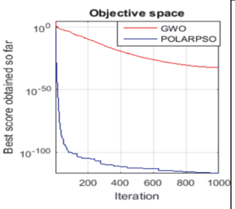
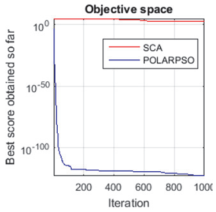
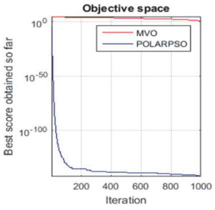
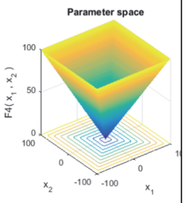
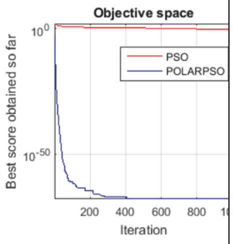
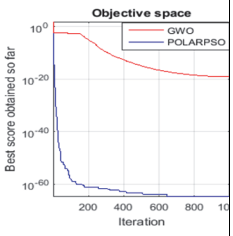
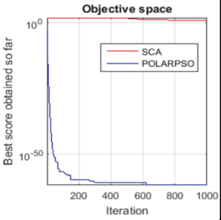
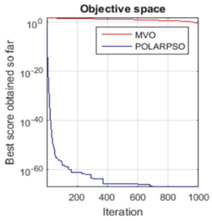
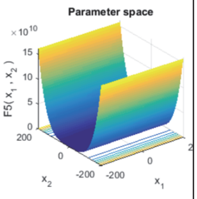
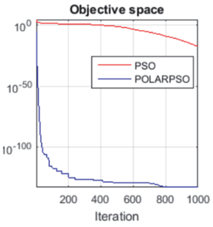
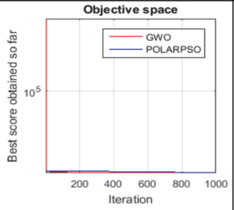
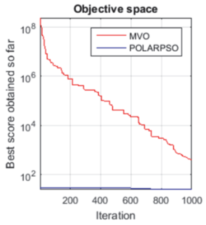
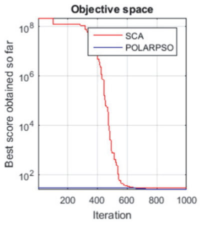
GRAPH	PSO	GWO	SCA	MVO
				

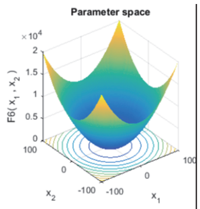
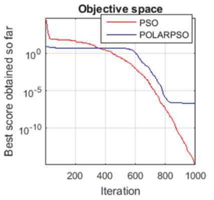
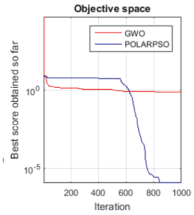
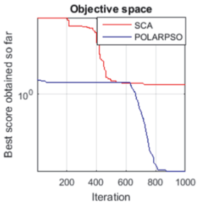
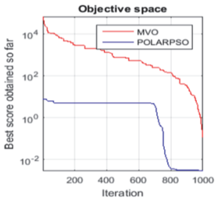
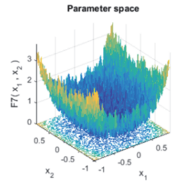
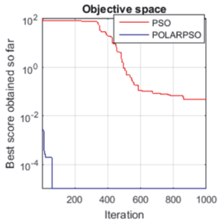
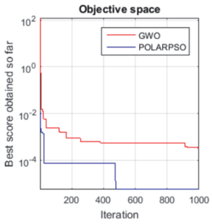
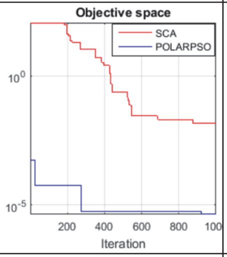
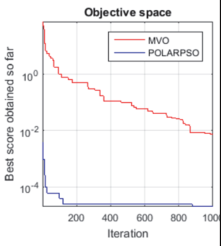
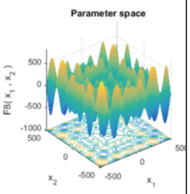
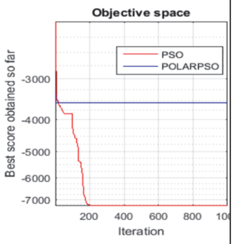
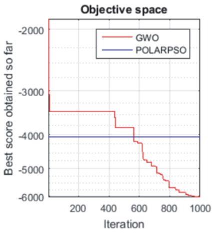
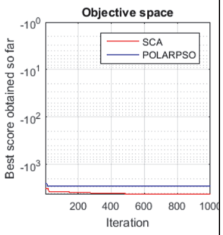
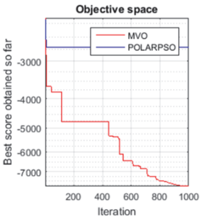
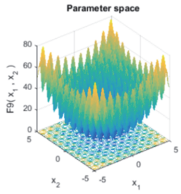
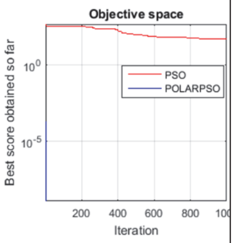
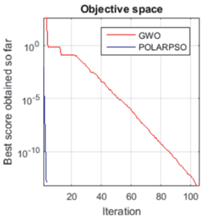
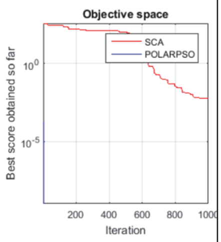
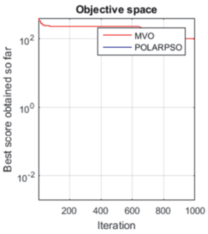
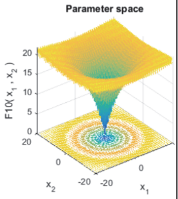
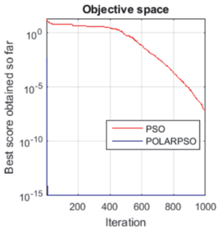
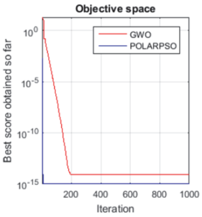
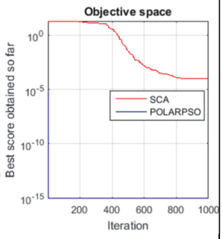
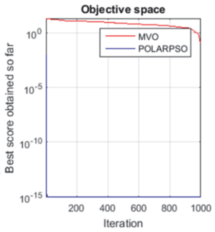
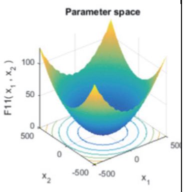
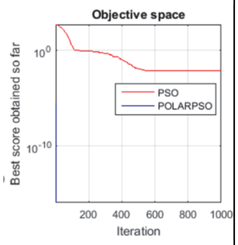
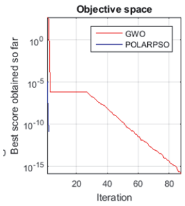
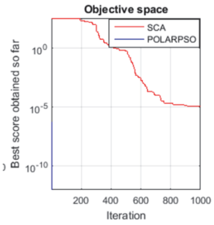
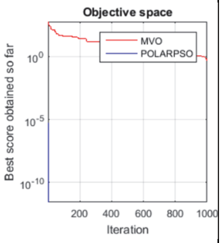
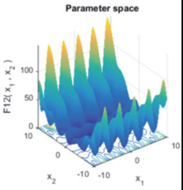
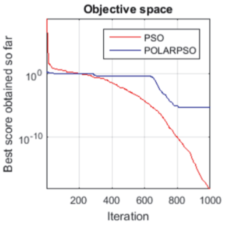
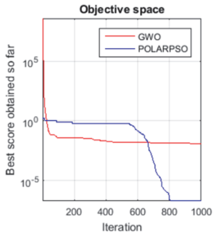
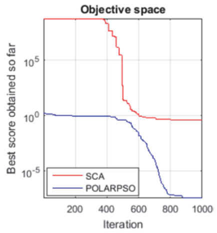
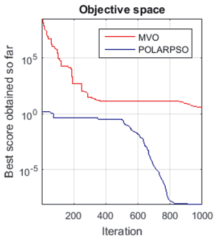
8. Convergence Analysis

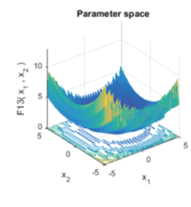
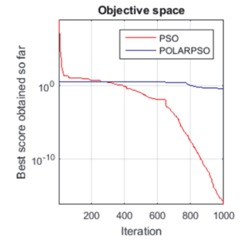
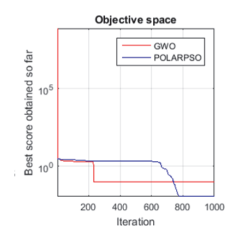
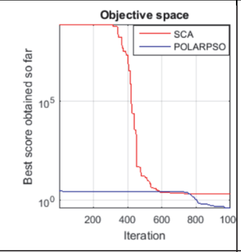
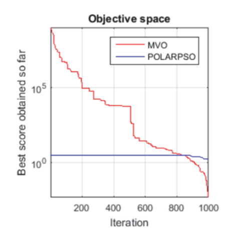
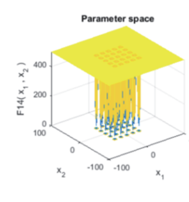
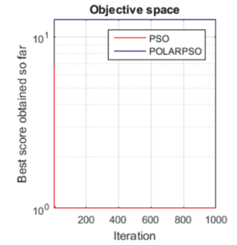
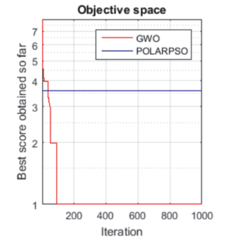
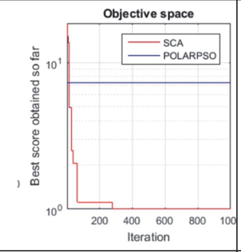
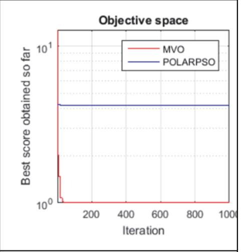
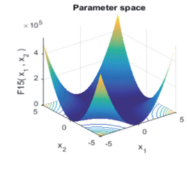
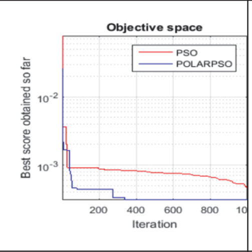
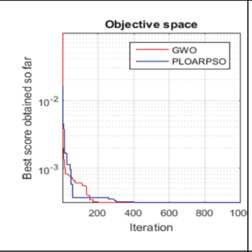
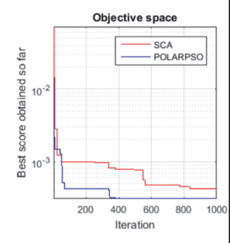
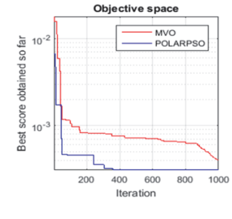
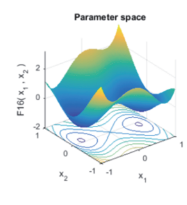
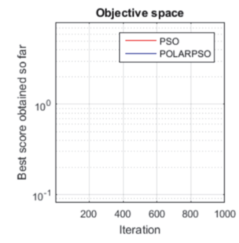
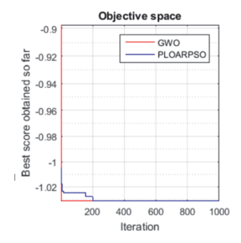
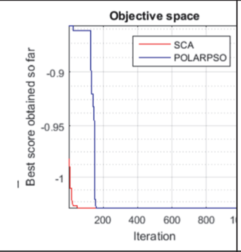
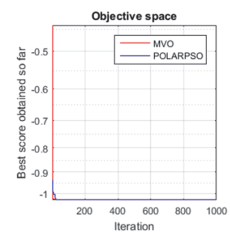
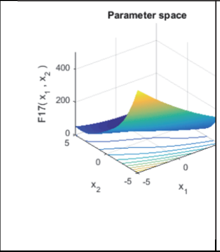
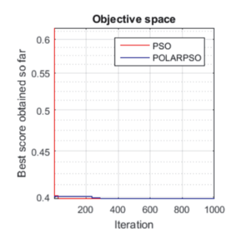
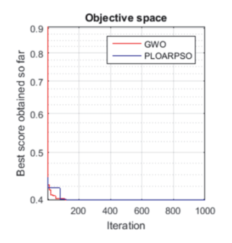
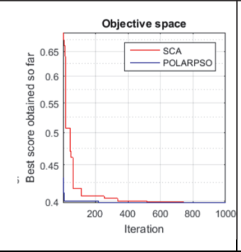
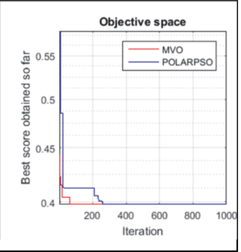
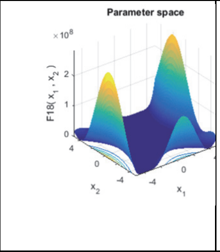
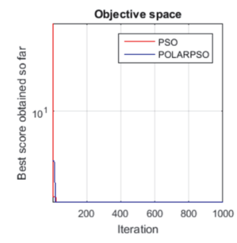
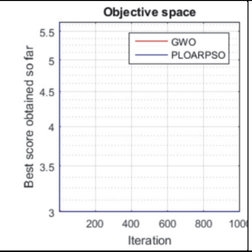
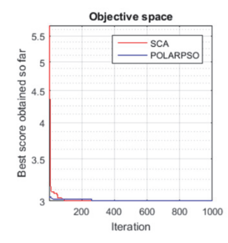
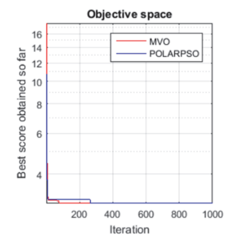
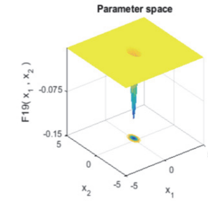
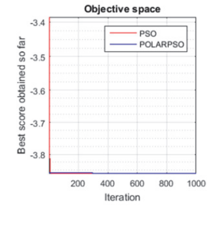
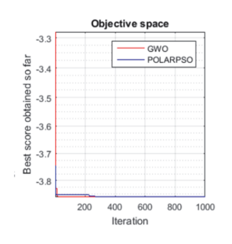
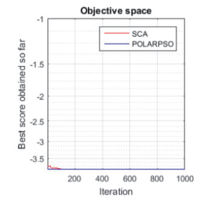
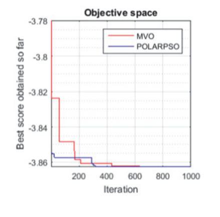
The POLARPSO showed a good solution for finding a good solution for the cloud data migration problem; this is due to fact that it keeps the best global minimum found so far and search for better one ahead. POLARPSO algorithm finds a better solution iteration by iteration.

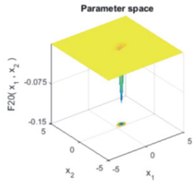
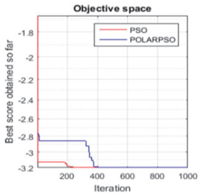
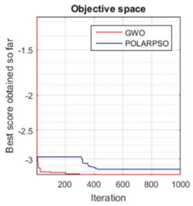
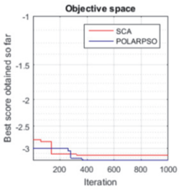
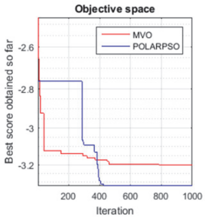
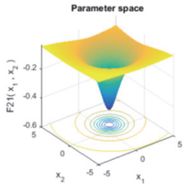
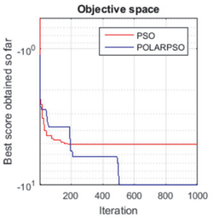
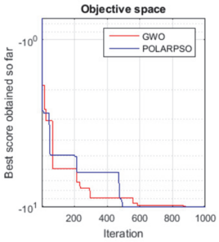
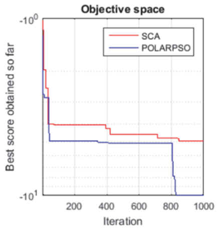
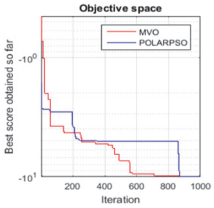
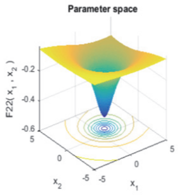
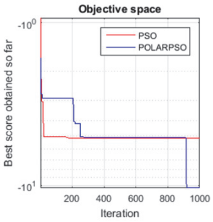
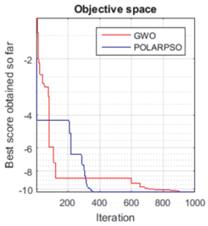
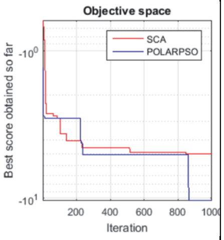
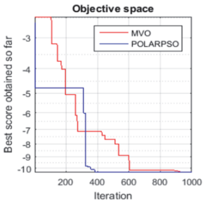
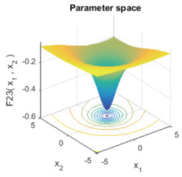
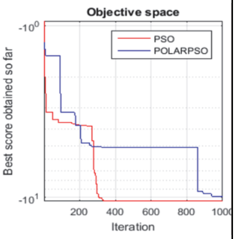
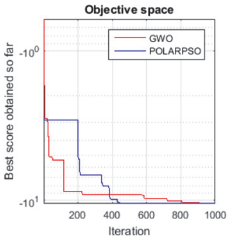
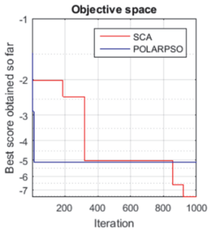
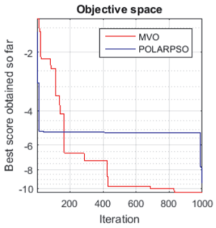
To test the speed of the POLARPSO in finding the optimal solution of the CEC2005 benchmark functions we drew the converge curve of POLARPSO compared to PSO, GWO, SCA and MVO. As shown in table 20. It can be seen from the table that the POLARPSO is very fast in finding the global minima for 20 functions out of 23 (function F1 through F7, F9 through F11, F15 through F23) when compared to all of PSO, GWO, SCA and MVO. In F12 it was faster than GWO, SCA and MVO.

Table 20. Converge curve for the cec2005 benchmark functions over POLARPSO compared to PASO, GWO, SCA and MVO

F	Graph	PSO	GWO	SCA	MVO
1					
2					
3					
4					
5					

6	 Parameter space F6(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration Best score obtained so far	 Objective space MVO POLARPSO Iteration Best score obtained so far
7	 Parameter space F7(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration	 Objective space MVO POLARPSO Iteration Best score obtained so far
8	 Parameter space F8(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration Best score obtained so far	 Objective space MVO POLARPSO Iteration Best score obtained so far
9	 Parameter space F9(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration Best score obtained so far	 Objective space MVO POLARPSO Iteration Best score obtained so far
10	 Parameter space F10(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration Best score obtained so far	 Objective space MVO POLARPSO Iteration Best score obtained so far
11	 Parameter space F11(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration Best score obtained so far	 Objective space MVO POLARPSO Iteration Best score obtained so far
12	 Parameter space F12(x1 , x2)	 Objective space PSO POLARPSO Iteration Best score obtained so far	 Objective space GWO POLARPSO Iteration Best score obtained so far	 Objective space SCA POLARPSO Iteration Best score obtained so far	 Objective space MVO POLARPSO Iteration Best score obtained so far

13					
14					
15					
16					
17					
18					
19					

20	 Parameter space F20(x₁ , x₂) x₁ x₂				
21	 Parameter space F21(x₁ , x₂) x₁ x₂				
22	 Parameter space F22(x₁ , x₂) x₁ x₂				
23	 Parameter space F23(x₁ , x₂) x₁ x₂				

9. Conclusion and Future Work

In this work, we proposed an enhanced optimization technique that is based on the well know PSO algorithm. We called the proposed algorithm POLARPSO. We tested our proposed algorithm on the CEC2005 benchmark set problems of 23 test functions classified in four categories: unimodal, multi-modal (basic and expanded) and the new hybrid composite functions. We compared the results recorded for POLARPSO with four state-of-the-art meta-heuristic optimization algorithms: PSO, GWO, SCA, and MVO. The results showed that POLARPSO outperformed the other meta-heuristic optimization algorithms.

We also considered using the proposed algorithm in solving a real-life problem of cloud data migration. Our proposed algorithm showed very competitive results and performed very well in solving this problem when compared with the other PSO, GWO, SCA, and MVO counterpart algorithms. This in turn motivates us to think about apply our proposed algorithm in other optimization problems such as cloud load balancing and scheduling.

References

Abbass, H. A. (2001). MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on* (Vol. 1, pp. 207-214). IEEE.

Askarzadeh, A., & Rezazadeh, A. (2013). A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer. *International Journal of Energy Research*, *37*(10), 1196-1204.

Binh, H. T. T. (2013). Hybrid particle swarm optimization for Solving multi-area economic dispatch Problem. *International Journal on Soft Computing*, *4*(2), 17.

Boyer, C. B. (1949). Newton as an originator of polar coordinates. *The American Mathematical Monthly*, *56*(2), 73-78.

Brown, R. G., Gleason, A. M., & Brown, M. A. (1992). *Advanced mathematics: Precalculus with discrete mathematics and data analysis*. Houghton Mifflin Company.

Campos, M., Bonabeau, E., Theraulaz, G., & Deneubourg, J. L. (2000). Dynamic scheduling and division of labor in social insects. *Adaptive Behavior*, *8*(2), 83-95.

Clerc, M. (2012). Standard particle swarm optimisation.

Coolidge, J. L. (1952). The origin of polar coordinates. *The American Mathematical Monthly*, *59*(2), 78-85.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39.

Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (pp. 39-43). IEEE.

Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, *17*(12), 4831-4845.

Gimmler, J., Stützle, T., & Exner, T. E. (2006, September). Hybrid particle swarm optimization: An examination of the influence of iterative improvement algorithms on performance. In *International Workshop on Ant Colony Optimization and Swarm Intelligence* (pp. 436-443). Springer Berlin Heidelberg.

Goldberg, E. D. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Adison Wesley PC.

Handl, J., & Meyer, B. (2007). Ant-based and swarm-based clustering. *Swarm Intelligence*, *1*(2), 95-113.

Introduction to the Polar Coordinate System. Boundless Algebra Boundless, 14 Oct. 2016. Retrieved 24 Jan. 2017 from https://www.boundless.com/algebra/textbooks/boundless-algebra-textbook/complex-numbers-and-polar-coordinates-344/the-polar-coordinate-system-417/introduction-to-the-polar-coordinate-system-418-17463/.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp.* 1942-1948.

Khanesar, M. A., Teshnehlab, M., & Shoorehdeli, M. A. (2007, June). A novel binary particle swarm optimization. In *Control & Automation, 2007. MED'07. Mediterranean Conference on* (pp. 1-6). IEEE.

Li, X. L. (2003). A new intelligent optimization-artificial fish swarm algorithm. *Doctor thesis, Zhejiang University of Zhejiang, China*.

Lu, X., & Zhou, Y. (2008, September). A novel global convergence algorithm: bee collecting pollen algorithm. In *International Conference on Intelligent Computing* (pp. 518-525). Springer Berlin Heidelberg.

Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120-133.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*(2), 495-513.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46-61.

Mucherino, A., & Seref, O. (2007, November). Monkey search: a novel metaheuristic search for global optimization. In O. Seref, O. E. Kundakcioglu, & P. Pardalos (Eds.), *AIP conference proceedings* (Vol. 953, No. 1, pp. 162-173). AIP.

Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, *26*, 69-74.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2005). The bees algorithm. Technical note. *Manufacturing Engineering Centre, Cardiff University, UK*, 1-57.

Pinto, P. C., Runkler, T. A., & Sousa, J. M. (2007, April). Wasp swarm algorithm for dynamic MAX-SAT problems. In *International Conference on Adaptive and Natural Computing Algorithms* (pp. 350-357).

Springer Berlin Heidelberg.

Roth, M. (2005). Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks.

Selvan, S. E., Subramanian, S., & Solomon, S. T. (2003, April). Novel technique for PID tuning by particle swarm optimization. In *Proc. 7th Annu. Swarm Users/Researchers Conf. (SwarmFest 2003)*.

Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009, May). A dolphin partner optimization. In *Intelligent Systems, 2009. GCIS'09. WRI Global Congress on* (Vol. 1, pp. 124-128). IEEE.

Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, *2*(2), 78-84.

Yang, X. S. (2011, May). Metaheuristic optimization: algorithm analysis and open problems. In *International Symposium on Experimental Algorithms* (pp. 21-32). Springer Berlin Heidelberg.

Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 210-214). IEEE.

Yu, S., Wu, Z., Wang, H., Chen, Z., & Zhong, H. (2012). A Hybrid Particle Swarm Optimization Algorithm Based on Space Transformation Search and a Modified Velocity Model. *International Journal of Numerical Analysis & Modeling*, *9*(2).

Yushui, G., & Jiaheng, Y. (2015, August). Cloud Data Migration Method Based on PSO Algorithm. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)* (pp. 143-146). IEEE.

Yushui, G., & Jiaheng, Y. (2015, August). Cloud Data Migration Method Based on PSO Algorithm. In Distributed Computing and Applications for Business Engineering and Science (DCABES), 201514th International Symposium on (pp. 143-146). IEEE.

Yushui, G., Jiaheng, Y., & Tao, S. (2016). Cloud Data Migration Method Based on ABC Algorithm. *International Journal of Database Theory and Application*, *9*(5), 141-148.

Copyrights