Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7– No. 23, December 2018 – www.caeaccess.org

Validation of Hybridized Particle Swarm Optimization (PSO) Algorithm with the Pheromone Mechanism of Ant Colony Optimization (ACO) using Standard Benchmark Function

Mogaji Stephen Alaba
Department of Computer Science
Federal University of Technology,
Akure, Nigeria

Alese Boniface Kayode
Department of Cyber Security Science
Federal University of Technology,
Akure, Nigeria

Adetunmbi Adebayo O
Department of Computer Science
Federal University of Technology,
Akure, Nigeria

ABSTRACT
Swarm intelligence (SI) is the communal behavior of devolved, self-organized structures, natural or artificial. SI systems consist typically of a population of simple agents interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents

This research work aims at hybridizing the conventional Particle Swarm Optimization (PSO) algorithm with the pheromone mechanism of Ant Colony Optimization (ACO) to attain faster convergence on a feasible standard PSO solution space then benchmarked against standard optimization test functions using Python Programming language to prove the correctness and convergence of the Hybridized PSO optimization mode for minimization. The result shows that hybridizing swarm intelligence performs better in solving difficult continuous optimization problems.

General Terms
Algorithm, Optimization, Algorithmic Pseudo-code, Swarm Intelligence,

Keywords
Hybridization, Pheromone mechanism, Benchmark functions, pbest, gbest, PSO, ACO

1. INTRODUCTION
Nature obscures many mysteries. In the past, behaviors like ants foraging and flight flocks were considered as magical secrets of nature. Nature always plays a vital role to solve complex human problems. In the past few years biology based techniques get the attentions of researchers in the field of Information Security. These and other phenomena inspired researchers to study and understand their secrets. The unraveling of many of these mysteries and secrets led to the foundation of new artificial intelligence science known as Swarm Intelligence (SI) [19].

A swarm is a large number of homogenous, simple agents interacting locally among themselves, and their environment, with no central control to allow a global interesting behavior to emerge. Swarm intelligence refers to systems, which accomplish complex global tasks through the simple local interactions of autonomous agents. The control is completely distributed among the individual agents with no leader coordinating any of the activities. Swarm intelligence is the emergent collective intelligence of groups of simple agents. It is a computational intelligence approach to solve real world complex problems.

Beni and Wang [6] first introduced it in cellular robotics system. Swarm intelligence systems buildup of a population of simple agents interactive with each other individually or with their environment. The inspiration of swarm intelligence comes from the biological or natural system. Insects, bees and birds in the form of swarms solve the complex problem that seems almost impossible at individual level.

Researchers have done so many works in this field and created many swarm intelligence based algorithms models and applications. Few important Swarm Intelligence (SI) algorithms are:

- i Ant colony optimization algorithm
- ii Artificial Bee colony algorithm
- iii Particle swarm optimization
- iv Firefly Algorithm
- v Multi-swarm optimization
- vi River Formation Dynamics
- vii Bacterial Foraging algorithm
- viii Cat Swarm Optimization algorithm
- ix Artificial Immune System algorithm
- x Glowworm Swarm Optimization algorithm

2. PARTICLE SWARM OPTIMISATION (PSO)
Particle swarm optimization (PSO) is a stochastic search technique considered as one of the modern heuristic algorithms for optimization, introduced by Kennedy and Eberhart [23]& [24]. It is based on the social behaviour metaphor of bird flocking and it is a population-based optimization technique.

According to Hazem and Janice [19]. The advent of flocking and schooling in assemblages of interacting agents (such as birds, fish, etc.) have long fascinated a wide range of scientists from diverse disciplines including animal behaviour, physics, social psychology, social science, and computer science for many years. Bird flocking can be defined as the social collective motion behaviour of a large number of interacting birds with a common group objective. The local collaborations among birds (particles) usually emerge the shared motion direction of the swarm. Such collaborations are based on the "nearest neighbour principle" where birds follow certain flocking rules to adjust their motion (i.e., position and velocity) based only on their nearest neighbours, without any central management The pioneering work of Reynolds [35] proposed three simple flocking rules to implement a simulated flocking behaviour of birds:

a. flock centering (flock members attempt to stay close to nearby flockmates by flying in a direction that keeps them closer to the centroid of the nearby flockmates),
b. Collision avoidance (flock members avoid collisions with nearby flockmates based on their relative position), and
c. Velocity matching (flock members attempt to match velocity with nearby flockmates).

In PSO, participant solutions of a population, called particles, coexist and evolve simultaneously based on knowledge sharing with neighbouring particles. While flying through the problem search space, each particle generates a solution using directed velocity vector. Each particle modifies its velocity to find a better solution (position) by applying its own flying experience (that is, memory having best position found in the earlier flights) and experience of neighbouring particles that is, best found solution of the population. Finally, all particles fly towards the best [35].

The standard PSO model consists of a swarm of particles, moving interactively through the feasible problem space to find new solutions. Each particle has a position represented by a position vector where ***n*** is the index of the particle and a velocity represented by a velocity vector. Each particle remembers its own best position so far in the vector, ***pbest*** and the best position vector among the swarm, ***gbest.***

The search for the optimal position (solution) advances as the particles' velocities and positions are updated. A particle's velocity and position are updated as follows:

$$v_{n+1} = wv_n + c_1 r_1 (p_{best} - x_n) + c_2 r_2 (g_{best} - x_n) \quad (1)$$
$$x_{n+1} = x_n + v_{n+1} \quad (2)$$

Where

- v_{n+1} = Velocity of the particle at n+1th iteration
- w = Particle inertia weight
- v_n = Velocity of particle at nth iteration
- c_1 = acceleration factor related to g_{best}, the cognitive scaling parameter
- c_2 = acceleration factor related to l_{best}, the social scaling parameter
- r_1 = random number between 0 and 1
- r_2 = random number between 0 and 1
- g_{best} = global best position on the swarm
- p_{best} = personal best position of the particle

The position of each particle in the swarm is affected both by the most optimist position during its movement (individual experience) and the position of the most optimist particle in its surrounding (near experience). Each solution vector can be confined to a vector range to control excessive roaming of particles outside the search space [19].

The particle weight inertial is reduced dynamically to decrease the search area in gradual fashion, using the equation below:

$$\mathrm{w} = (\mathrm{w}_{\max} - \mathrm{w}_{\min}) \times \frac{(\mathrm{t}_{\max} - \mathrm{t})}{\mathrm{t}_{\max}} + \mathrm{w}_{\min} \quad (3)$$

- $\mathrm{w}_{\max}$ = Maximum particle weight inertia
- $\mathrm{w}_{\min}$ = Minimum particle weight inertia
- $\mathrm{t}_{\max}$ = Given maximum number of iterations

Particle flies toward a new position using equation (1) and (2). All particles of the swarm find their new positions and apply these new positions to update their individual best position and global best position of the swarm. This process is repeated until maximum number of iteration count $\mathrm{t}_{\max}$ is reached.

```
BEGIN Algorithm
1. Initialise, search space (P) and maximum epoch (t_max)
2. Initialise random particles positions and velocities
   # initialise optimisation
3. For each particle in the search space
4. Calculate the fitness value (by evaluate the objective function)
5. If the fitness value (pbest) is better than the best fitness value in history
6. Set current value as the new pbest
7. End If
8. Find the particle with best (min) fitness value of all particles as the gbest
9. End For
   # perform optimisation
10.    While (t <= maximum epoch)
11.    Update particle velocity according to the particle velocity equation
12.    Update particle position according to position update equation
       Calculate the fitness value by evaluating the objective function using current
       velocity
       Update particle best (min) by comparing previous fitness and current fitness,
       as pbest
13.    Find the best (min) fitness value of all particles as gbest
14.    Increment epoch count t+=1
15.    End While
       Report the best solution gbest of the swarm having already evaluated the
       objective function
END Algorithm
```

Figure 1 PSO Algorithmic Pseudo-Code

3. ANT COLONY OPTIMISATION (ACO)

In the 1990's, Ant Colony Optimization was introduced as a novel nature inspired method for the solution of hard optimization problems [12].

Ants, like many other social insects, communicate with each other using volatile chemical substances known as pheromones, whose direction and intensity can be perceived with their long,

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7– No. 23, December 2018 – www.caeaccess.org

mobile antennae. The term "pheromone" was first introduced by Karlson and Lüscher [22], based on the Greek word pherein (means to transport) and hormone (means to stimulate). There are different types of pheromones used by social insects. One example of pheromone types is alarm pheromone that crushed ants produce as an alert to nearby ants to fight or escape dangerous predators and to protect their colony [28].

Another important type of pheromone is food trail. Ants live on the ground and make use of the soil surface to leave pheromone trails, which can be followed by other ants on their way to search for food sources. Ants that happened to pick the shortest route to food will be the fastest to return to the nest, and will reinforce this shortest route by depositing food trail pheromone on their way back to the nest.

The inspiring source of ACO is the foraging behaviour of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates it and carries some food back to the nest. During the return trip, the ant deposits a pheromone trail on the ground. The pheromone deposited, the amount of which may depend on the quantity and quality of the food, guides other ants to the food source. As it has been shown [17], indirect communication among ants via pheromone trails enables them to find shortest paths between their nest and food sources.

In the ACO algorithm, an artificial ant colony simulates the pheromone trail following behaviour of real ants. Artificial ants move on a construction graph representing a specific problem to construct solutions successively. The artificial pheromone that corresponds to the record of routes taken by the ant colony is accumulated at run-time through a learning mechanism. Individual ants concurrently collect necessary information, stochastically make their own decisions and independently construct solutions in a stepwise manner. The information required for making a decision at each step includes pheromone concentration, problem related data and heuristic function values. The pheromone laid on the path belonging to the iteration-best solution will be positively increased to become more attractive in the subsequent iterations. Because of self-organize and reverse-engineering behaviour, ACO can effectively and efficiently solve a wide class of combinatorial optimization problems. This capability of real ant colonies has inspired the definition of artificial ant colonies that can find approximate solutions to hard multimodal optimization problems. The central component of ACO algorithms is the pheromone model, which is used to probabilistically sample the search space. Recently, there are few adaptations of ACO for solution of continuous optimization problems. In this work, a simple pheromone-guided search mechanism of ant colony is implemented which acts locally to synchronize positions of the particles of PSO to quickly attain the feasible domain of objective function.

4. SYSTEM ANALYSIS AND DESIGN

4.1 Hybridization of Particle Swarm Optimisation using Ant Colony Optimisation

Swarm intelligence meta-heuristics, namely, particle swarm optimisation and ant colony optimisation are proven to be successful approaches to solve complex optimization problems. PSO algorithm, whose concept began as a simulation of a simplified social environment, is a powerful optimization technique for solving multimodal optimization problems [33], [8] & [32]. ACO imitates foraging behaviour of real life ants, and are known to be efficient and robust for solution of combinatorial optimization problems [42], [11], [50], & [46].

```
BEGIN Algorithm
1. Initialise, ACO and PSO design variables, search space (P) and maximum epoch (t_max)
2. Initialise random particles positions and velocities
# initialise optimisation
3. For each particle in the search space
4. Calculate the fitness value (by evaluate the objective function)
5. If the fitness value (pbest) is better than the best fitness value in history
a. Set current value as the new pbest
6. End If
7. Find the particle with best (min) fitness value of all particles as the gbest
8. End For
# perform optimisation
9. While (t <= maximum epoch)
10.     Update particle velocity according to the particle velocity equation
11.     Update particle position according to position update equation
12.     Calculate the fitness value by evaluating the objective function
13.     Update particle best (min) by comparing previous and current fitness, as pbest
14.     Find the best (min) fitness value of all particles as gbest
15.     Generate P solutions, z, from gbest value according to zsolution equation
16.     Calculate fitness value by evaluating objective function on generated z solution
17.     Update particle best position by comparing zsolution and particle objective solution
18.     Find the best (min) fitness value of all particles pbest and gPbest, as gbest
19.     Increment epoch count t=t+1
20.     End While
21.     Report the best solution gbest of the swarm
END Algorithm
```

Figure 2 (Hybridized PSO) Model Algorithmic Pseudo-code

The implementation of this proposed algorithm comes in two stages. In the first stage, PSO is applied while ACO is implemented in the second stage. ACO works as a local search, wherein, ants apply pheromone-guided mechanism to update the positions found by the particles in the earlier stage, to attain rapid convergence on a feasible solution space. The implementation of ACO in the second stage of this model is based on the studies of Angeline [8] which shows that:

i. PSO discovers reasonable quality solutions much faster than other evolutionary algorithms
ii. If the swarm is going to be in equilibrium, the evolution process will be stagnated as time goes on. Thus, PSO does not possess the ability to improve upon the quality of the solutions as the number of generations is increased.

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7– No. 23, December 2018 – www.caeaccess.org

In this proposed model, a simple pheromone-guided mechanism of ACO is proposed to apply as local search.

The proposed ACO algorithm handles P ants equal to the number of particles in PSO. Each ant i generate a solution $\boldsymbol{z_t}$ around $\boldsymbol{g_{best}}$ the global best-found position among all particles in the swarm up to iteration count t as [43]:

$$z_t = N(g_{best}, \sigma) \quad (4)$$

The components of the solution vector z_t which satisfies the Gaussian distribution with mean g_{best} and standard deviation σ is generated, where, initially at t = 1 value of σ = 1 and is updated at the end of each iteration as

$$\sigma = \sigma \times d \quad (5)$$

d is a parameter in (0.25, 0.997) and if $\sigma < \sigma_{min}$ then $\sigma = \sigma_{min}$, where, σ_{min} is a parameter in (10^{-2} , 10^{-4}).

The objective function around z_t, $f(z_t)$ is the computed and replaces the current position of the particle swarm if $f(z_t) < f(x_t)$ the $x_t = z_t$

This simple pheromone-guided mechanism considers there is highest density of trails (single pheromone spot) at the global best solution ***gbest*** of the swarm at any iteration t + 1 in each stage of ACO implementation and all ants P search for better solutions in the neighbourhood of the global best solution. In the beginning of the search process, ants explore larger search area in the neighborhood of ***gbest*** due to the high value of standard deviation r and intensify the search around ***gbest*** as the algorithm progresses [43]. ACO pheromone mechanism helps PSO process, not only to efficiently perform global exploration for rapidly attaining the feasible solution space, but also to effectively reach optimal or near optimal solution [42]. The (Hybridized PSO) Model Algorithmic Pseudo-code and the flowchart are shown in Figure 2 and Figure 3 respectively

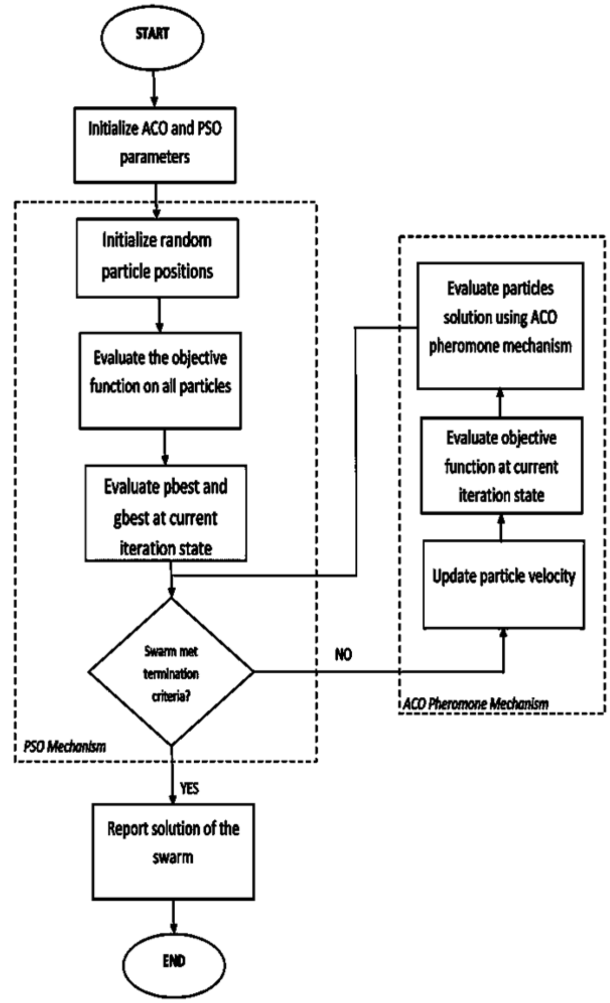

Figure 3 Flowchart diagram for proposed hybridized PSO with ACO

5. IMPLEMENTATION AND RESULTS

This section presents the implementation and report of the proposed hybridized model tested and validated against three benchmark functions namely Ackley function, Rastrigin function and Rosenbrock function to evaluate the optimization of the hybridized algorithm and compare result with standard PSO.

A python program was developed to test the proposed model against these benchmark functions.

The hybridized PSO algorithm and standard PSO algorithm is experimented on 50 particles in solution space over 100 iterations. The results of the validation are present below.

5.1 Ackley Fitness Function

The Ackley function, proposed by David Ackley [1] is an n-dimensional function that has a large number of local minima but only one global minimum. It is a typical problem to solve

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7– No. 23, December 2018 – www.caeaccess.org

with evolutionary algorithms and is widely used for testing optimization algorithms.

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e \quad (6)$$

5.2 Rastrigin Fitness Function

Rastrigin function, proposed by Rastrigin [34], is a non-convex function used as a performance test problem for optimization algorithms. It is a typical example of non-linear multimodal function and has many local minima and one global minimum. The farther the local minimum is from the origin, the larger the value of the function is at that point.

$$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 + 10\cos(2\pi x_i)] \quad (7)$$

5.3 Rosenbrock Fitness Function

In mathematical optimization, the Rosenbrock function is a non-convex function, introduced by Howard H. Rosenbrock [36], which is used as a performance test problem for optimization algorithms. It is also known as Rosenbrock's valley or Rosenbrock's banana function

$$f(x) = (1-x)^2 + 10(y - x^2)^2 \quad (8)$$

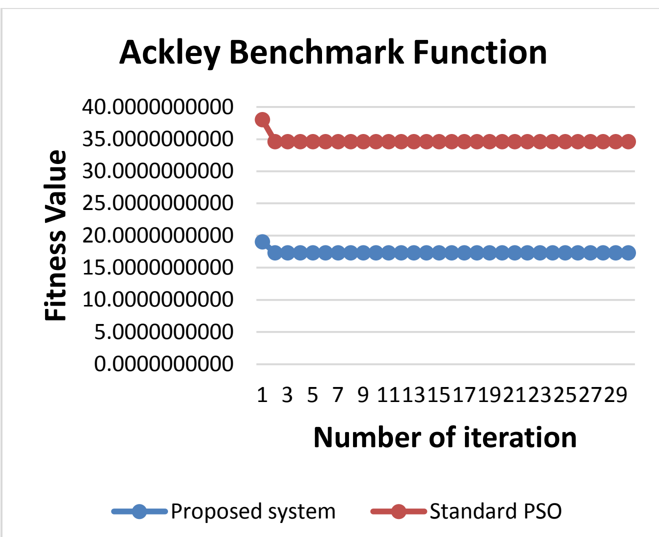

Figure 4: Ackley benchmark function against Standard PSO and Hybridized PSO after 100 iterations.

Table 1: Ackley benchmark function against Standard PSO and Hybridized PSO after 100 iterations.

Iteration	Hybridized PSO	Standard PSO
10	17.2926113508	17.2919271035
14	17.2926113233	17.2919272349
15	**17.2926113232**	17.2919272349
20	17.2926113232	17.2919271035
25	17.2926113232	17.2919271029
26	17.2926113232	**17.2919271028**
30	17.2926113232	17.2919271028
40	17.2926113232	17.2919271028
50	17.2926113232	17.2919271028
60	17.2926113232	17.2919271028
70	17.2926113232	17.2919271028
80	17.2926113232	17.2919271028
90	17.2926113232	17.2919271028
100	17.2926113232	17.2919271028

From the data Figure 4 and Table 1, it is noted that using the Hybridized PSO function, the Ackley benchmark solution converges on the 15th iteration, in contrast to the standard PSO algorithm that converges on the 26th iteration, representing a 42.31% increase in convergence speed.

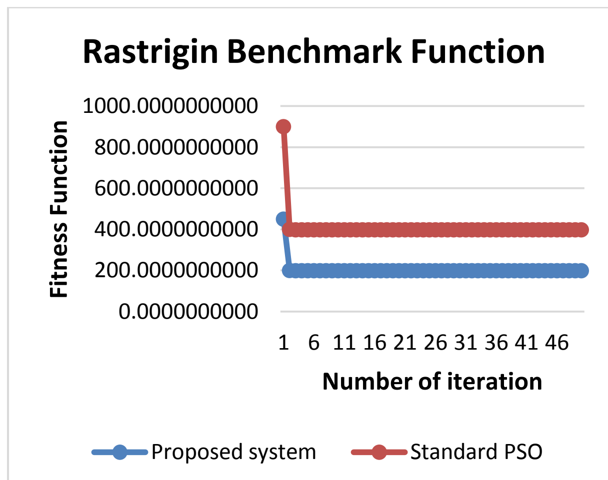

Figure 5: Rastrigin benchmark function against Standard PSO and Hybridized PSO after 50 iterations

Table 2: Rastrigin benchmark function against Standard PSO and Hybridized PSO after 100 iterations

Iteration	Hybridized PSO	Standard PSO
10	198.9837645750	198.9832800765
20	198.9832499365	198.9832501278
29	198.9832488068	198.9832488295
30	**198.9832488064**	198.9832488295
40	198.9832488064	198.9832488065
41	198.9832488064	198.9832488065
42	198.9832488064	**198.9832488064**
50	198.9832488064	198.9832488064

60	198.9832488064	198.9832488064
70	198.9832488064	198.9832488064
80	198.9832488064	198.9832488064
90	198.9832488064	198.9832488064
100	198.9832488064	198.9832488064

The table 2 and figure 5 above reflects the superiority of the Hybridized PSO optimization model over standard PSO model. It is shown that using the proposed hybridized model, convergence is achieved upon the 30th iteration count. This reflects a 28.57% convergence speed increase when compared to the standard PSO model, which attains convergence upon the 42nd iteration, when solving the Rastrigin benchmark function.

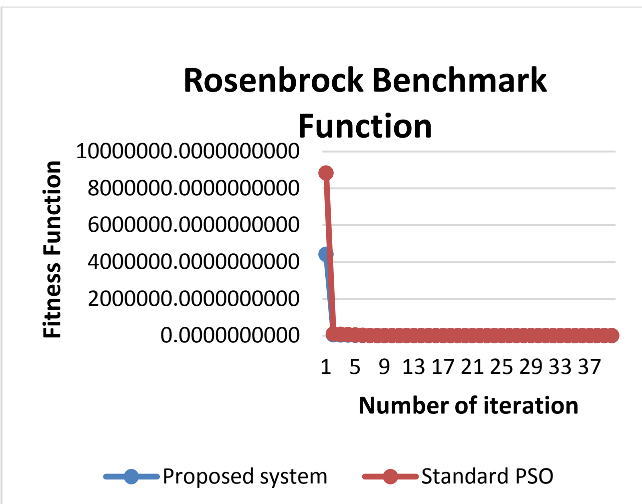

Figure 6: Rosenbrock benchmark function against Standard PSO and Hybridized PSO after 40 iterations.

Table 3: Rosenbrock benchmark function against Standard PSO and Hybridized PSO after 100 iterations

Iteration	Hybridized PSO	Standard PSO
10	1.6041532037	1.594699936
20	1.5937445326	1.593740284
27	1.5937382194	1.593740284
28	**1.5937381959**	1.593740284
30	1.5937381959	1.593738207
37	1.5937381959	1.593738197
38	1.5937381959	**1.593738196**
40	1.5937381959	1.593738196
50	1.5937381959	1.593738196
60	1.5937381959	1.593738196
70	1.5937381959	1.593738196
80	1.5937381959	1.593738196
90	1.5937381959	1.593738196
100	1.5937381959	1.593738196

Figure 6 and Table 3 data further strengthens the Hybridized PSO model's advantage over standard PSO model. It is shown that using the Hybridized PSO model, convergence is achieved upon the 28th iteration count. This reflects a 26.32% convergence speed increase when compared to the standard PSO model, which attains convergence upon the 38th iteration, when solving the Rosenbrock benchmark function.

6. CONCLUSIONS

The proposed hybridized model was tested and validated against three well-known optimization benchmark functions namely Ackley function, Rastrigin function and Rosenbrock function, using 50 particles in solution space over 100 iterations. Against Ackley fitness function, a 42.31% improvement is convergence speed is identified when compared to existing standard PSO model, 28.57% when tested against Rastrigin benchmark function and 26.32% improvement using the Rosenbrock function. The comparison of the numerical result of the hybridized PSO with standard PSO shows that hybridizing swarm intelligence is better in solving difficult continuous optimization problems.

7. REFERENCES

[1] Ackley, D. H. (1987) "A connectionist machine for genetic hill-climbing", Kluwer Academic Publishers, Boston MA.

[2] Angeline, P.J. (1998). Evolutionary optimization versus particle swarm optimization: philosophy and performance difference, in: V.W. Porto et al. (Eds.), Proceedings of 7th Annual Conference on Evolutionary Programming, Lecture Notes in Computer Science, vol. 1447, Springer, Berlin, 1998, pp. 601–610.

[3] Amudha, P., and Abdul Rauf, H. (2012). A Study on Swarm Intelligence Techniques in Intrusion Detection, IJCA Special Issue on Computational Intelligence & Information Security

[4] Bao, F., Chen I.R., Chang M., and Cho, J.H. (2011). Hierarchical trust management for wireless sensor networks and its application to trust-based routing. Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 1732- 1738.

[5] Bao, F., Chen, R., Chang, M., and Cho, J. H. (2012). Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, Network and service Management, IEEE Transactions on, vol. 9, pp. 169-183.

[6] Beni, G., W.J. (1989). Swarm intelligence in cellular robotics systems: NATO Advanced Workshop on Robots and Biological System.

[7] Bin, Y., Zhong-Zhen, Y., and Baozhen, Y. (2009). An improved ant colony optimization for vehicle routing sensor networks, IEEE Communications Surveys & Tutorials, pp: 2-28.

[8] Dong, Y., Tang, J., Xu, B., and Wang D., (2005). An application of swarm optimization to nonlinear programming, Computers & Mathematics with Applications 49 (11–12) pp1655–1668.

[9] Dong, P., Wang H., and Zhang H. (2009). Probability-based trust management model for distributed e-commerce, Network Infrastructure and Digital Content. IC-NIDC. IEEE International Conference, pp. 419-423.

[10] Dorigo, M. (1992). Optimization, Learning and Natural Algorithms (in Italian). Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[11] Dorigo, M., Blum C. (2005). Ant colony optimization theory: a survey, Theoretical Computer Science 344 (2–3) (2005) 243–278

[12] Dorigo, M., Maniezzo, V., and Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics - Part B 26(1):29–41

[13] Dorigo, M., Gambardella, L.M. (1997). Ant colony system: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation

[14] Dorigo, M., Di Caro, G. (1999). The ant colony optimization meta-heuristic, McGraw-Hill Ltd.,UK, Maidenhead, UK, England, pp 11–32 1:53–66

[15] Dorigo, M., Caro, G.D., Gambardella, L.M. (1999). Ant algorithms for discrete optimization. Artificial Life 5(2):137–172

[16] Dorigo, M,. and Stützle, T. (2004). Ant Colony Optimization. MIT Press, Cambridge, ISBN: 978-0-262-04219-2.

[17] Goss, S., Aron, S., Deneubourg, J., and Pasteels, J. (1989). Self-Organized Shortcuts in the Argentine Ant, Naturwissenchaften, Vol. 76, pp. 579-581,

[18] Hao, Y., et al. (2004). Security in mobile ad hoc networks: challenges and solutions. Wireless Communications, IEEE, 11(1): p. 38-47.

[19] Hazem, A., and Janice, G. (2012). Swarm Intelligence: Concepts, Models and Applications. Technical Report School of Computing Queen's University Ontario Canada.

[20] He, Q., Wu, D., and Khosla, (2004) P. "SORI: A Secure and Objective Reputation-based Incentive Scheme for Ad-Hoc Networks," Proc. IEEE Wireless Communications and Networking Conf., vol. 2, pp. 825- 830.

[21] Iftikhar, M.S., and Fraz, M.R. (2013). A Survey on Application of Swarm Intelligence in Network Security in Transactions on Machine Learning and Artificial Intelligence, Volume 1, No 1, PP 01-15.

[22] Karlson, P., and Lüscher, M. (1959). Pheromones: a new term for a class of biologically active substances. Nature, Vol. 183, pp. 55–56.

[23] Kennedy, J., and Eberhart, R.C. (1995). Particle swarm optimization. In Neural Networks, 1995. Proceedings. IEEE International Conference on. IEEE.

[24] Kennedy, J., and Eberhart, R.C (1995). Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948.

[25] Kennedy, J., and Eberhart, R.C (1995). A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43.

[26] Li, W., Parker, J., and Joshi A. (2012). Security through collaboration and trust in MANETs, Mobile Networks and Applications, vol. 17, pp. 342-352.

[27] Li, R., and Li, J. (2013). Requirements and design for neutral trust management framework in unstructured

[28] Lloyd, C. (2003). The alarm pheromones of social insects: A review, Technical report, Colorado State University.

[29] Luo, J., Liu, X., and Fan, M. (2009). A trust model based on fuzzy recommendation for mobile ad-hoc networks. Computer Networks, vol. 53, pp. 2396-2407.

[30] Nagalakshmi, S., and Rakesh, P. (2016). Performance Comparison and Evaluation of Efficient Routing Protocols for MANETs: Ant Inspired Adaptive Routing. *International Journal of Advanced Research in Computer and Communication Engineering* ISO 3297:2007 Certified Vol. 5, Issue 9.

[31] Nekkantim R.K., and Lee, C. (2004) "Trust-based Adaptive On Demand Ad Hoc Routing Protocol," Proc. 42th Annual ACM Southeast Regional Conf., Huntsville, Alabama, 2004, pp. 88-93.

[32] Ourique, C.O., Biscaia, E.C., and Pinto, J.C.(2002). The use of particle swarm optimization for dynamical analysis in chemical processes, Computers & Chemical Engineering 26 (12) 1783–1793.

[33] Paterlini, S., and Krink, T. (2006). Differential evolution and particle swarm optimisation in partitional clustering, Computational Statistics & Data Analysis" 50 (5) 1220–1247.

[34] Rastrigin, L. A. (1974) "Systems of extremal control." Mir, Moscow.

[35] Reynolds, C.W. (1987). Flocks, herds, and schools: A distributed behavioural model, Computer Graphics (ACM SIGGRAPH '87 Conference Proceedings), Vol. 21, No. 4, pp. 25–34.

[36] Rosenbrock, H.H. (1960). "An automatic method for finding the greatest or least value of a function". The Computer Journal. 3 (3): 175–184. doi:10.1093/comjnl/3.3.175. ISSN 0010-4620

[37] Shabut, A.R.M. (2015). Trust Computational Models for Mobile Ad Hoc Networks, a Ph.D. thesis submitted to School of Computing Faculty of Engineering and Informatics University of Bradford

[38] Shabut, A., Dahal, K.P., and Awan, I. (2013). A Recommendation-Based Trust Model for MANETs to Enhance Dynamic Recommender Selection Using Multiple Rules, Seventh International Open Conference HET-NETs 2013, UK, Ilkely.

[39] Shabut, A., Dahal, K.P., Awan, I. (2013). A Trust-Based Monitoring Model to Secure Routing Protocol in MANETs Using Enhanced Trust Metric", Seventh International Open Conference HET-NETs 2013, UK, Ilkely, 2013.

[40] Shabut, A., and Dahal, K.P. (2011). Trust and Security Management in Distributed Systems, University of Bradford school of computing, School Research seminars.

[41] Sharvani G.S. (2012) "Development of Swarm Intelligent Systems for MANET" A Ph.D Thesis submitted to the Department of Computer Science and Engineering, Faculty of Engineering, Avinashilingam University for Women, Coimbatore

[42] Shelokar, P.S., Jayaraman, V.K., and Kulkarni, B.D. (2004). An ant colony classifier system: application to some process engineering problems. Computers & Chemical Engineering 28 (9) (2004) 1577–1584

[43] Shelokar, P., Siarry, P., Jayaraman, V. K., & Kulkarni, B. D. (2007). Particle swarm and ant colony algorithms hybridized for improved continuous optimization. Applied Mathematics and Computation, 188(1), 129–142.

[44] Shi, Y (2004) "Feature article on particle swarm optimization", IEEE Neural Network Society, Feature Article, pp. 8–13.

[45] Shyu, S.J., Yin, P.Y., Lin, B.M.T. (2004). An ant colony optimization algorithm for the minimum weight vertex cover problem. Annals of Operations Research 131:283–304

[46] Shyu, S.J., Lin, B.M.T., and Hsiao, T.S. (2006). Ant colony optimization for the cell assignment problem in PCS networks, Computers & Operations Research 33 (6) (2006) 1713–1740.

[47] Wang Y.D., and Emurian, H .H. (2005). An overview of online trust: Concepts, elements, and implications, Computers in human behavior, vol. 21, pp. 105-125, 2005.

[48] Wang, Y.F., Hori, Y., and Sakurai, K. (2008). Characterizing economic and social properties of trust and reputation systems in P2P environment, *Journal of Computer Science and Technology*, vol. 23, pp. 129-140, 2008.

[49] Wang, Y. and Vassileva, J. (2003). Trust and reputation model in peer-to-peer networks. Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on, 2003, pp. 150-157

[50] Yin, P.Y., and Wang, J.Y., (2006). Ant colony optimization for the nonlinear resource allocation problem. Applied Mathematics & Computation 174 (2)

[51] Yunfang, F (2007) "Adaptive Trust Management in MANETs," Proc. 2007 Int'l Conf. on Computational Intelligence and Security, Harbin, China, pp 804-808.