ISSN: 2633-4828 Vol. 5 No.3, September, 2023

International Journal of Applied Engineering & Technology

HYBRID PSOABC ALGORITHM FOR ACHIEVING OPTIMUM FITNESS VALUE IN CLOUD ENVIRONMENT

Er Beant Singh[1]
Lecturer Computer Engineering
Government Polytechnic College, Bathinda
Department of Technical Education and Industrial Training, Punjab

Dr. Rajbhupinder Kaur[2]
Assistant Professor in Computer Science and Engineering
Yadavindra department of engineering, Punjabi University Guru Kashi Campus, Talwandi Sabo, Bathinda, Punjab

Dr. Seema[3]
Assistant Professor in Computer Science and Engineering
Yadavindra department of engineering, Punjabi University Guru Kashi Campus, Talwandi Sabo, Bathinda, Punjab

Dr. Manoj Kumar[4]
Assistant Professor in Computer Science and Engineering
Yadavindra department of engineering, Punjabi University Guru Kashi Campus, Talwandi Sabo, Bathinda, Punjab

Corresponding Author: **Dr. Rajbhupinder Kaur**

ABSTRACT
Cloud computing is an evolution of parallel, distributed, cluster, and grid computing. The rapid growth of storage and processing technologies in computing resources is more powerful and cheaper than conventional computing. Recently there has been an upsurge in the use of Swarm Intelligence algorithms for different research purposes in cloud computing. The research paper elaborates on the functioning of two popular swarm intelligence algorithms: Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC). The working of both PSO and ABC is elaborated with a detailed algorithm and flowchart. Both algorithms are intended to find the best cost value for multiple iterations using the best cost functions. Multiple parameters like decision variables, a maximum number of iterations, population size, inertia coefficient, damping ratio, and personal and social acceleration coefficient have been considered for PSO. The parameters considered for ABC are decision variables, the maximum number of iterations, population size, number of onlooker Bees, abandonment limit parameter, and acceleration coefficient. The research paper implemented both algorithms in 25 iterations. The best cost value for PSO dropped from 10.9327 to 2.1745 and for ABC it dropped from 32.8435 to 0.14177. The obtained results for both PSO and ABC illustrate the continuous fall in the value of the best cost which indicates the positive performance of both algorithms. The research paper proposes a Hybrid PSOABC algorithm. The primary aim of conducting the research in the paper is to find the optimal path from source to destination at optimum fitness value. The hybrid PSOABC algorithm is constituted from the composition of PSO and ABC algorithms intended to find the optimal path between source and destination avoiding obstacles. The fitness value has been evaluated for each iteration using the fitness function along with the violation factor that occurred.

Keywords: *Artificial Bee Colony, Best Cost, Fitness value, Particle Swarm Optimization, Swarm Intelligence.*

I. INTRODUCTION
The concept of SI (Swarm Intelligence) has been derived from the ability of animals to collectively handle problems. The working of SI is based on the principle of self-organization among lower-level components to create a global-level dynamic structure (Tong et al., 2020). This is regarded as intelligence. The individuals follow the set of rules to initiate the interactions at the lower level without analyzing the effects of these interactions

globally in advance. Individuals contain information at the local level (Afzal & Kavitha, 2019). These local-level interactions utilize direct/indirect methods of communication and affect the global organization of the colony. Optimization algorithms have a primary goal to search for the optimal solutions following handling any existing problems subject to a given set of constraints (Batista et al., 2019). Several computational models have evolved based on studies conducted on social animals having relevance to swarm intelligence (Yiqiu et al., 2019). The cooperative behavior within these swarms is quite complex. Some of the prominent optimization methods designed based on the biology of animals or insects are PSO (Particle Swarm Optimization), ABC (Artificial Bee Colony), GA (Genetic Algorithm), and ACO (Ant Colony Optimization). The working of SI is based on the principle of self-organization among lower-level components to create a global-level dynamic structure (Jairam Naik, 2020). The individuals follow the set of rules to initiate the interactions at the lower level without analyzing the effects of these interactions globally in advance (C. Zhang et al., 2019). Individuals contain information at the local level. Optimization algorithms have a primary goal to search for the optimal solutions following handling any existing problems subject to a given set of constraints (Janakiraman & Ramya, 2019).

The fitness function is intended to evaluate the nearness of the given solution to the optimum solution for any given problem. There is no hard and fast rule that a specific problem should have a specific function. The fitness function and the process of calculating the fitness score should be clear to the users. The fitness function should be such that it should not become the bottleneck of the algorithm. It should not hinder the overall efficiency of the swarm algorithm. The function should quantitatively measure the fitness of the solution in solving the problem. The function should provide instinctive results. The best contenders should have the best score and the worst should have the worst. The swarm algorithms signify each solution as a string of binary numbers called a chromosome. One has to assess these solutions and find out the best set of solutions for cracking a given problem. The evaluation is performed by providing scores to each solution to designate its nearness to the general specification of the desired solution. The score is generated by smearing the fitness function to the test.

The research paper comprises multiple sections. Section II illustrates the studied literature. The research papers comprising the research work conducted on different swarm algorithms, job scheduling and load balancing, energy consumption, and calculating best cost values have been studied and analyzed. Section III elaborates on the adopted research methodology. This section focuses on the working of the PSO and ABC intended to calculate the best cost value. The algorithms and detailed flowcharts for both the swarm intelligence algorithms are mentioned in the section. Thereafter, the proposed Hybrid PSOABC algorithm has been described with a descriptive algorithm and flowchart. The proposed algorithm inherits the properties of both PSO and ABC and worked on finding the best path from source to destination with the best fitness value. Section IV implements the instance for each algorithm. Multiple parameters have been considered and results have been obtained. The obtained readings have been shown in the respective tables of each algorithm. Finally, Section V winds up the research paper with appropriate concluding remarks.

II. STATE-OF-THE-ART

Jeyafzam et al., 2021 described that in medical science, collecting and classifying data from various diseases is a vital task. Diabetes diagnosis relies on large amounts of data with many parameters, it is necessary to use machine learning methods such as SVM (Support Vector Machine) to predict the complications of diabetes. One of the disadvantages of SVM is its parameter adjustment, which can be accomplished using metaheuristic algorithms such as PSO, GA, or GWO. SVM is used to predict complications of diabetes based on selected parameters of a patient acquired by laboratory tests using improved GWO. The anticipated method outperforms the other conventional techniques. *Arulkumar & Bhalaji, 2021* stated that Cloud computing is an on-demand service. Cloud computing comprises data centers and user bases. Data centers are referred to the servers and user bases to the clients. The primary concern will the focus needs to be laid is job scheduling. The job of job scheduling is to balance the load among available resources to achieve optimized performance as per the client's request. This could be further enhanced by selecting a data center that is closer to the client for providing service to the client. The authors analyzed the performance of nature-inspired algorithms meant for job scheduling and load balancing like PSO, ACO, and WWA (Water Wave Algorithm) based on total response time and data processing time. The cloud analyst has been used to carry out the simulation and the authors concluded that WWA is best in terms of total response time and PSO in terms of data processing time. *Farid et al., 2020* stated that cloud computing comprises a network of servers that are placed in remote areas free of geographical boundaries. Cloud computing makes use of workflow models to represent different web and scientific applications. The major issue is scheduling workflows of processes in a heterogeneous cloud environment. The user can be satisfied with the service provided by cloud computing in the terms of parameters like scalability, reliability, effectiveness, and maximized end-user resource utilization. The authors analyzed the scheduling algorithms based on the performance of PSO. Moreover, scheduling schemes are classified according to the variant of the PSO algorithm applied. The authors highlighted the objectives, features, and limitations found. Finally, further directions for future research are identified. *Siddiqui et al., 2019* stated that alongside the numerous advantages of cloud computing, there are several challenges accompanied. The authors discussed job scheduling and load balancing algorithms like RR (Round Robin), Min - Min, Max-Min, OLB (Opportunistic Load Balancing) algorithm, GA (Genetic Algorithm), ABC, ACO, FireFly, and Cuckoo search algorithms based on parameters like response time, throughput, resource utilization, cost overhead, and performance. The authors concluded that the Cuckoo search algorithm performs better than other algorithms discussed in the research paper. *Xiao et al., 2019* stated that ABC is a strong and powerful optimization method with a strong ability to search. However, studies have also proved that ABC is not that efficient in the case of complex optimization problems. The paper proposes an improved ABC variant based on elite strategy and dimension learning (ABC-ESDL). The proposed algorithm is capable of searching for better solutions. In the trials, a classical benchmark set and the 2013 IEEE Congress on Evolutionary (CEC 2013) benchmark set are tested. Computational results show the planned ABC-ESDL accomplishes more precise solutions than ABC and five other improved ABC variants. *Ullah et al., 2019* stated that cloud computing has attracted a lot of attention in the world of information and technology. The primary objective of job scheduling and load balancing is to minimize the energy consumed and maximize resource consumption. Swarm Intelligence is preferred in cases where the problem is hectic to be handled and requires the involvement of classical mathematical techniques. ABC is an algorithm having robustness, high flexibility, strength, and fast convergence. The paper is a review paper focusing on the use of the ABC algorithm in job scheduling and load balancing in the cloud environment. *X. Zhang & Hu, 2019* proposed a mathematical example for a multi-constrained routing optimization problem. The multi-objective optimization problem is converted into a single-objective optimization problem by introducing a penalty function. ABC algorithm is utilized to search for the best route. The chances of the ABC algorithm falling into local optimal deficiencies are high, so the dynFWA (Dynamic Fireworks Algorithm) is constructed and used for local search which ensures fast guarantee and global search. The success rate is upgraded by 1.05% when compared with the PSO_ACO algorithm enhanced by the Ant Colony algorithm which is 6.18% higher than the standard PSO algorithm and the standard ABC algorithm. The minimum average cost of the search is about 0.53% more than the PSO_ACO algorithm, which is about 1.87% higher than the other two algorithms. *Ahmad & Khan, 2017* stated that cloud computing shares available resources and applications. The major challenge faced by cloud computing is job scheduling and load balancing. Load balancing involves managing network load, storage load, and load on processors. Efficient job scheduling and load balancing enable improvement in the performance of the network and build a fault-tolerant system that can be modified as per requirements. Different algorithms exist for job scheduling and load balancing in cloud computing each with its advantages and disadvantages. The authors performed experiments on SJF (Shortest Job First), RR (Round Robin), HBA (Honey Bee Algorithm), and ACO (Ant Colony Optimization) and concluded that the performance of the HBA is best in terms of load balancing. *Yuce et al., 2013* stated that Optimization algorithms are a kind of each problem where the main aim is to find the optimal path as per the available set of constraints. Swarm intelligence has evolved from the studies conducted on social animals and insects. The structure of such swarms is too complex. Two prominent algorithms of Swarm Intelligence are the Particle Swarm Optimization and the Ant-Colony algorithm. The paper describes the working of the Honey Bee algorithm for finding the optimized solution to a problem. The algorithm works on both an exploitative neighborhood and a random explorative search. The paper provides clarification regarding the natural searching behavior possessed by honey bees and its enhanced versions are analyzed to upgrade multiple benchmark functions.

III. RESEARCH METHODOLOGY

This section elaborates on the working of PSO and ABC algorithms using comprehensive algorithms in Fig. 1 and Fig. 2 respectively. Both PSO and ABC have been hybridized and the proposed methodology for the Hybrid PSOABC algorithm has been illustrated via an extensive flowchart in Fig. 3.

PSO

In PSO, random solutions and searches are used to initialize targets via updating generations. Particles fly through the problem space by following the current optimum particles. Each particle moving in the solution space is fascinated by two poles, its past best position (solution) and the best position (solution) of the whole swarm (collection of particles). These poles are responsible for transforming the velocity vector of the particles at each iteration. Algorithm 1 depicts the functioning of PSO.

Algorithm 1

- Initiate by initializing the decision variables (*nVar*).
- Set the matrix size (*VarSize*), lower bound (*VarMin*), and upper bound (*VarMax*) of the decision variables.
- Set the values for number of iterations (*MaxIt*), population size (*nPop*), damping ratio of inertia coefficient (*wdamp*), personal coefficient (*c1*) and social acceleration coefficient (*c2*).
- Initialize the particle template and create the population array.
- Initialize the global best and the population members.
- Generate random solution and initialize velocity.
- Update the personal best and the global best and the cluster head.
- Initialize array to hold best cost value on each iteration and evaluate the energy dissipated by the Cluster head to non-cluster heads.
- Store the best cost value and display the iterations information as output with graphical representation.

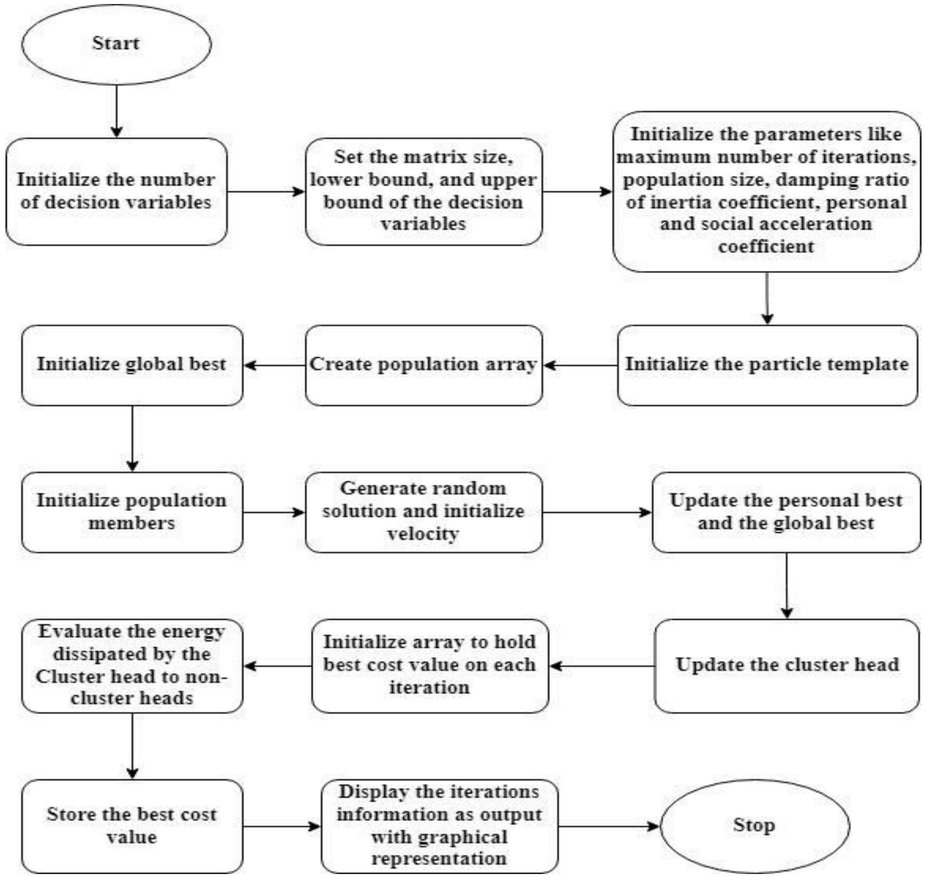


Fig. 1 Flowchart depicting the working of PSO

ABC

The Bee-inspired algorithm is a meta-heuristic swarm intelligence algorithm based on the foraging behavior of honeybee colonies to solve numerical optimization problems. The Artificial Bee Colony algorithm comprises three groups of bees. These are classified as employed bees, onlookers, and scouts.

Algorithm 2

- Initiate setting up a desired cost function.
- Initialize number and matrix size of decision variables (*VarSize*), lower bound (*VarMin*) and upper bound (*VarMax*) of the decision variables.
- Set the values for the variables like maximum number of iterations (*MaxIt*), population size (*nPop*), number of onlooker bees (*nOnlooker*), abandonment limit parameter (*L*), acceleration coefficient of upper bound (*a*).
- Empty bee structure and initialize population array and initialize best solution ever found. Create initial population and abandonment counter.
- Employee bees and declare acceleration coefficient. Find a new bee position and apply bounds. Evaluate and perform comparison.
- Calculate fitness values and selection probabilities and convert cost to fitness.
- Select source site for onlooker bees and declare acceleration coefficient. Set new bee position and apply bounds.
- Consider scout bees and update best solution ever found. Store best cost ever found and display iteration information.

The assumption is made that there exists only one artificially employed bee for each food source. Employed bees are responsible for going to the food source and returning to their hive and dancing in this area. On looker's bees watch the employed bees dancing and select their food sources depending upon the dances of the employed bees. The employed bees whose food source has been abandoned turn to scout and search for a new food source. This behavior is needed to recruit intelligent forager bees to find rich food sources, resulting in positive feedback. Similarly, the rejection of poor sources of food by foragers causes negative feedback. Algorithm 2 depicts the working of the ABC algorithm.

The flowchart in Fig. 2 depicts the working of ABC mentioned in Algorithm 2.

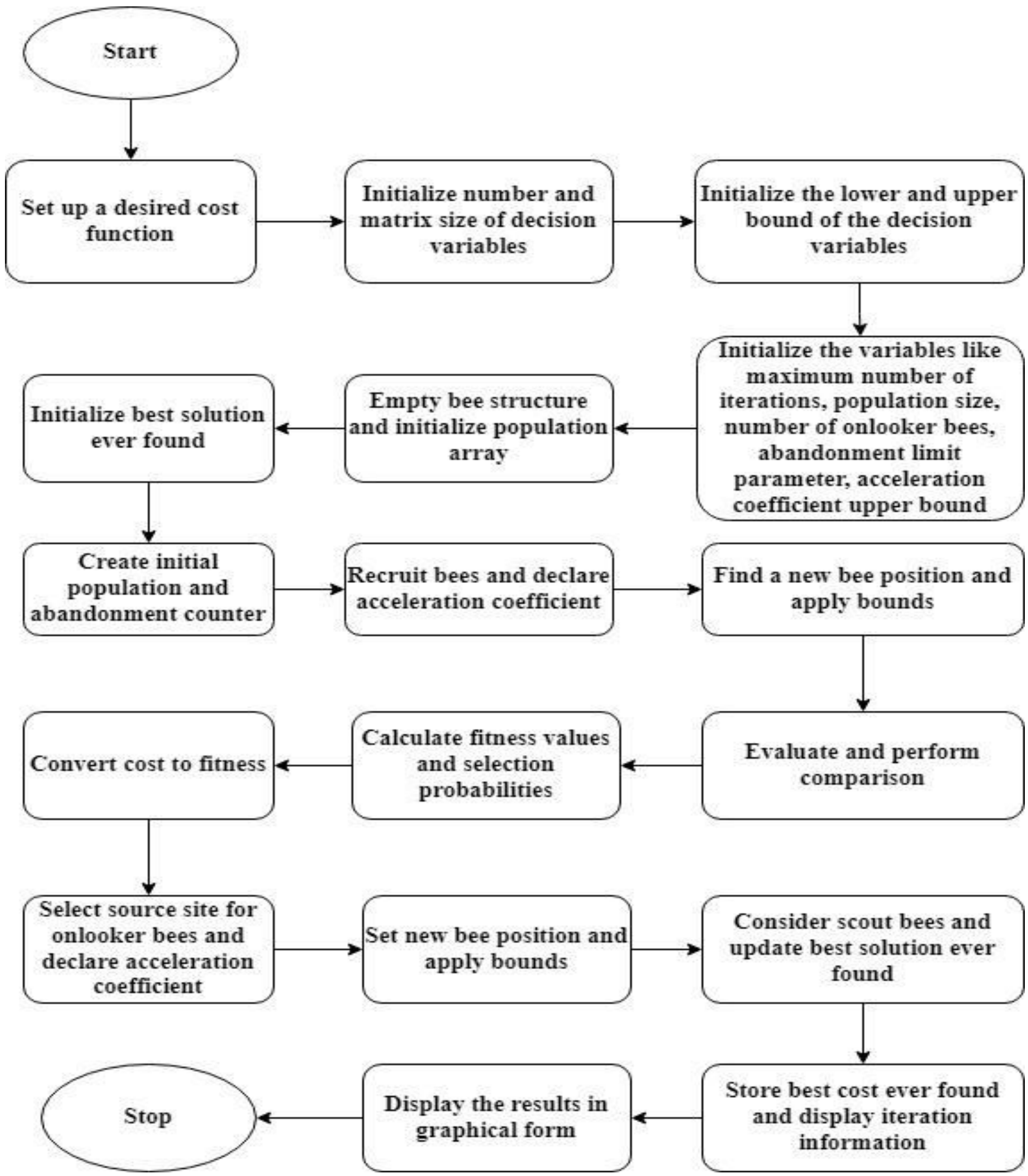


Fig. 2 Methodology adopted for executing ABC algorithm

Hybrid PSOABC

There is always a scope for improvement in different disciplines to find the best possible solutions. Solving problems with effective calculations and functions is critical for improvement. Hybrid algorithms have a noteworthy role in enhancing the searchability of algorithms, reducing the best cost values, finding the optimum path from the source to the destination, reducing energy consumption, and using the available resources to the optimum. The advantages of multiple algorithms are combined into a single algorithm minimizing the substantial disadvantages to form a hybrid algorithm. The hybridized algorithm is an improved version and scores in terms of computational speed, accuracy, etc. The proposed algorithm comprising the features of PSO and ABC to find the optimal path with the best fitness value from source to destination is depicted in Algorithm 3.

Algorithm 3

- Initialize the number of handle points (*model.n*) and the cost function, the number and size of decision variables (*nVar*).
- Initialize the lower bound (*VarMin.x* and *VarMin.y*) and upper bound (*VarMax.x* and *VarMax.y*) of the variables along with other parameters.
- Create Empty Particle Structure and empty Bee Structure and initialize the global best and population array.
- Initialize Best Solution Ever Found. Create Particles Matrix. Initialize positions and draw a straight line from source to destination.
- Initialize velocity and evaluate the cost function. Update personal best and the global best. Initialize array to Hold Best Cost Values at Each Iteration.
- Recruit Bees and define acceleration coefficients. Set up new Bee position and apply bounds to perform evaluation. Calculate Fitness Values and Selection Probabilities.
- Convert Cost to Fitness. Select Source Site for Onlooker Bees. Define Acceleration Coefficient and new bee position and apply bounds for evaluation. Update Best Cost Ever Found.
- Initialize Scout Bees and find the best solution ever found. Update velocity, velocity bounds, position, and position bounds.
- Evaluate and calculate personal and global best. Calculate Inertia Weight Damping and display the iterations information.

The flowchart in Fig. 3 depicts the proposed methodology for the Hybrid PSOABC algorithm discussed in Algorithm 3.

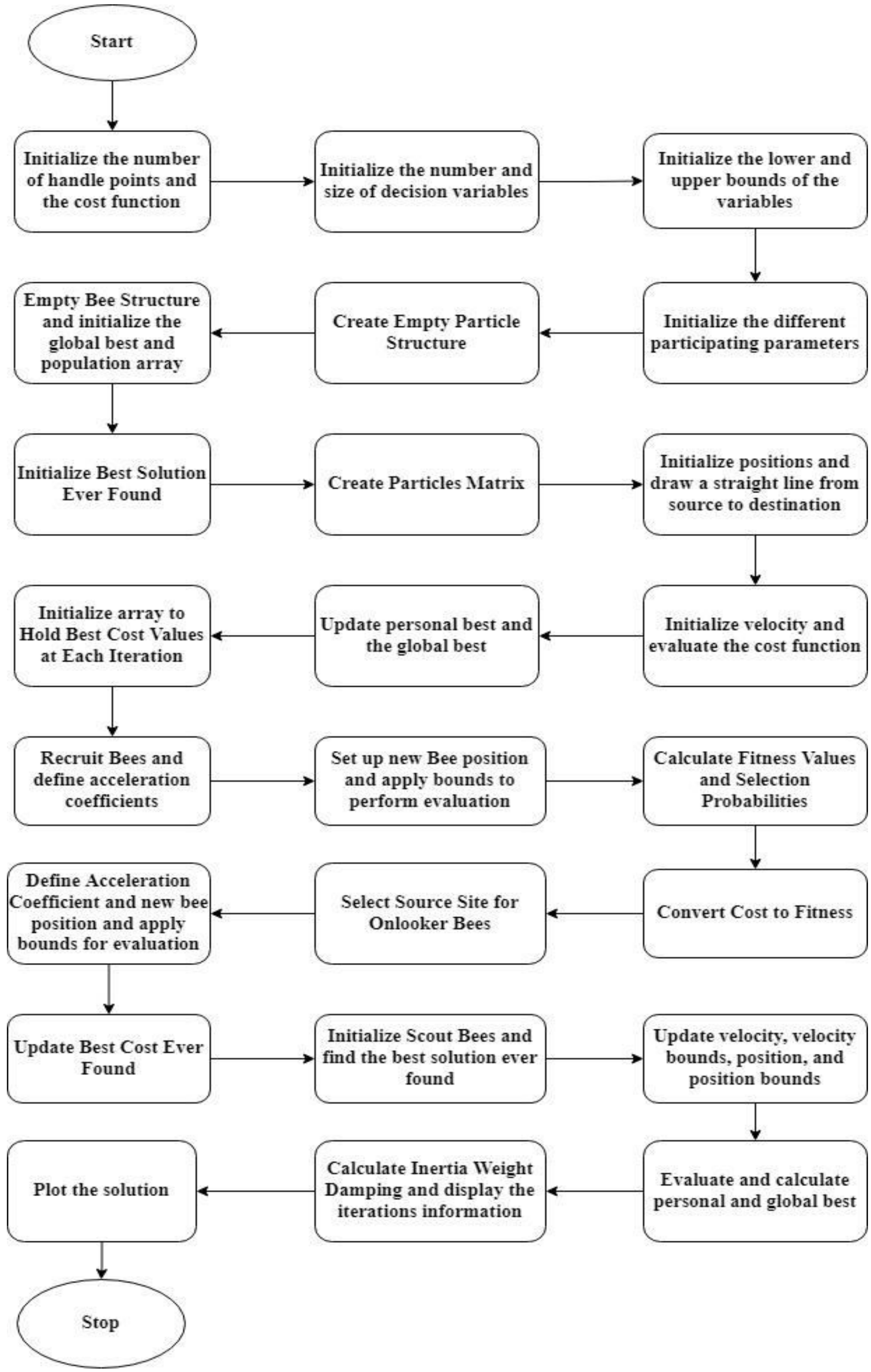


Fig. 3 Methodology adopted for executing Hybrid PSOABC algorithm

IV. IMPLEMENTATION AND RESULTS

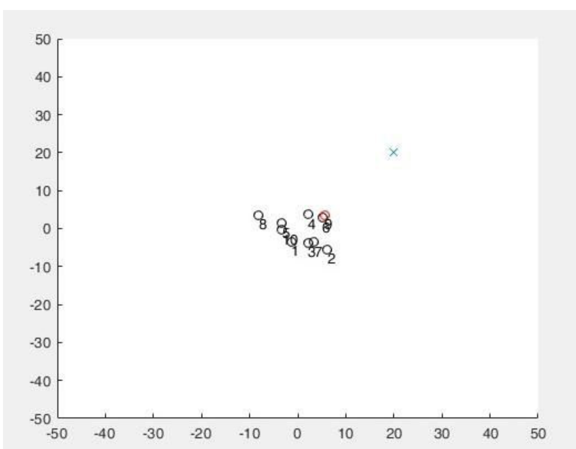
This section comprises three instances demonstrating the execution of the PSO, ABC, and the proposed algorithm respectively.

PSO

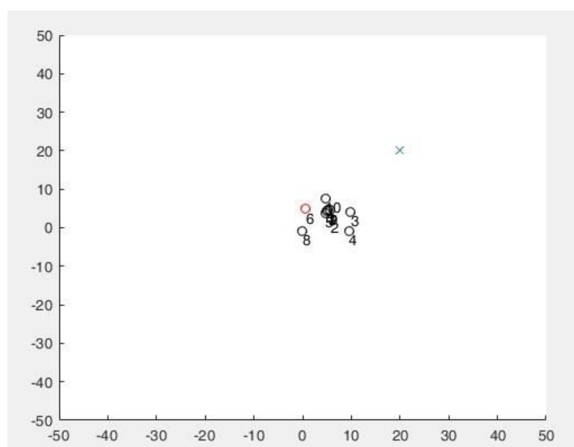
The different parameters considered with assigned values to execute the working of PSO are given below:

- Number of decision variables (*nVar*)= 2
- Matrix size of decision varibales (*VarSize*) = [1 nVar]
- Lower bound of decision varibales (*VarMin*) = -10
- Upper bound of decision variables (*VarMax*) = 10
- Maximum number of iterations (*MaxIt*) = 25
- Population Size (*nPop*) = 10
- Intertia Coefficient (*w*) = 1
- Damping Ratio of Inertia Coefficient (*wdamp*) = 0.99
- Personal Acceleration Coefficient (*c1*) = 2
- Social Acceleration Coefficient (*c2*) = 2

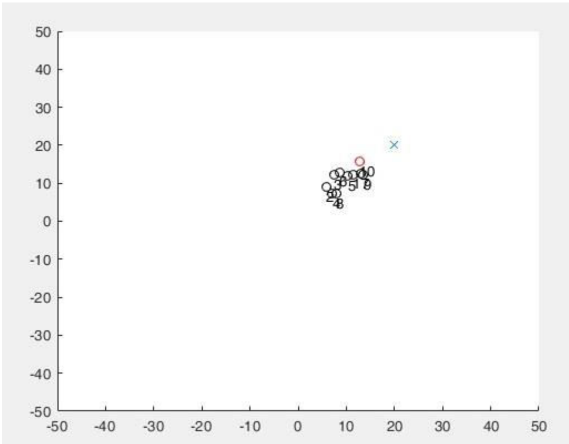
Fig. 4 shows the movements of the swarms as per the methodology defined in the flowchart mentioned in Fig. 1 to achieve the best cost.



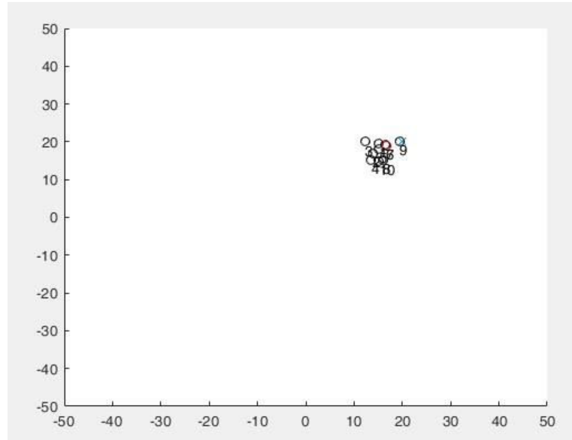
(1)



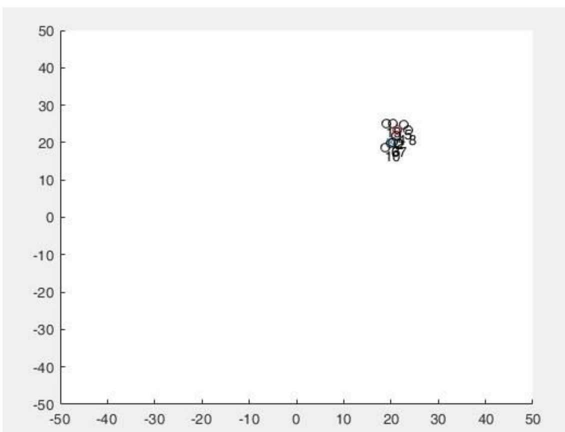
(2)


50 40 30 20 10 0 -10 -20 -30 -40 -50
-50 -40 -30 -20 -10 0 10 20 30 40 50

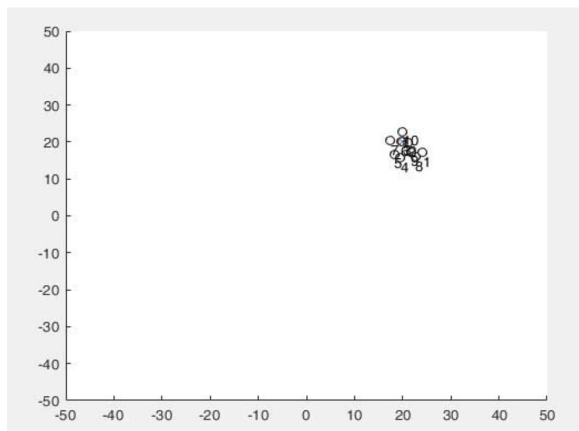
(3)


50 40 30 20 10 0 -10 -20 -30 -40 -50
-50 -40 -30 -20 -10 0 10 20 30 40 50

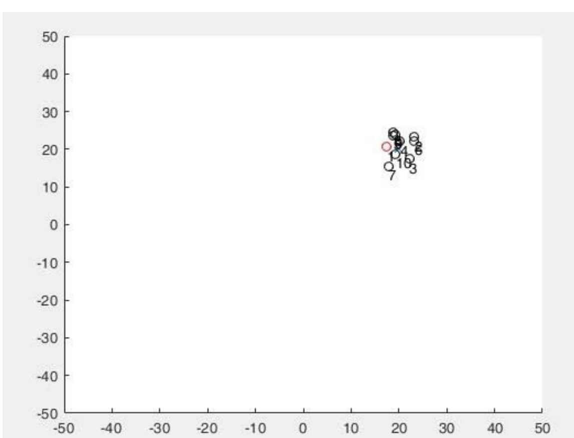
(4)


50 40 30 20 10 0 -10 -20 -30 -40 -50
-50 -40 -30 -20 -10 0 10 20 30 40 50

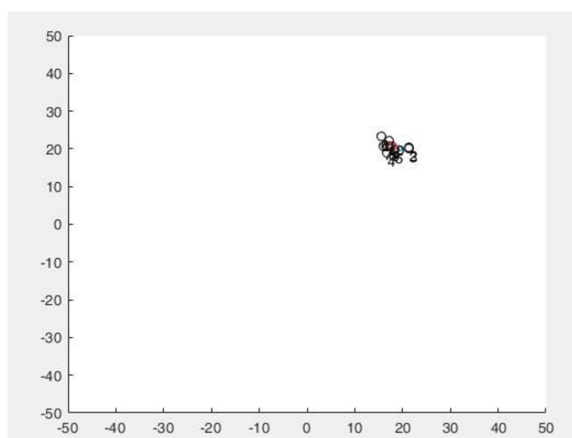
(5)


50 40 30 20 10 0 -10 -20 -30 -40 -50
-50 -40 -30 -20 -10 0 10 20 30 40 50

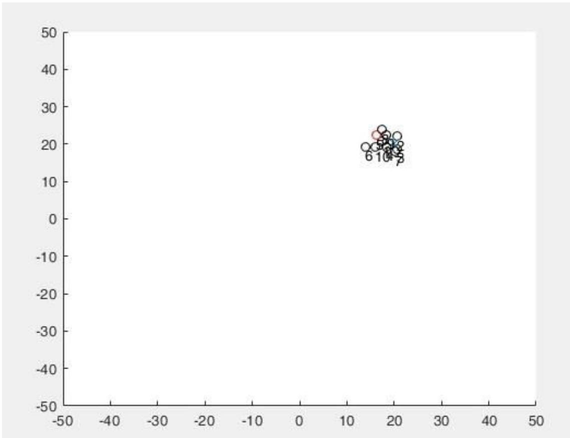
(6)


50 40 30 20 10 0 -10 -20 -30 -40 -50
-50 -40 -30 -20 -10 0 10 20 30 40 50

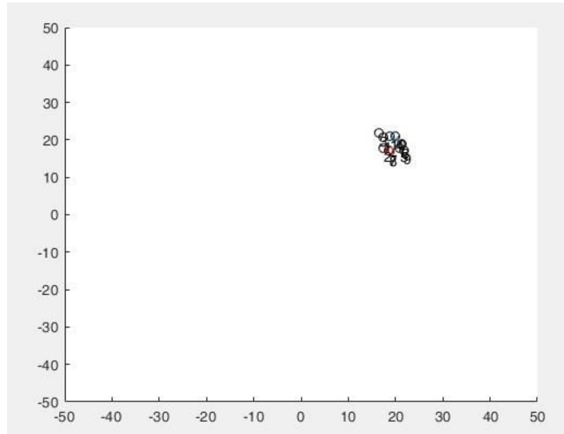
(7)


50 40 30 20 10 0 -10 -20 -30 -40 -50
-50 -40 -30 -20 -10 0 10 20 30 40 50

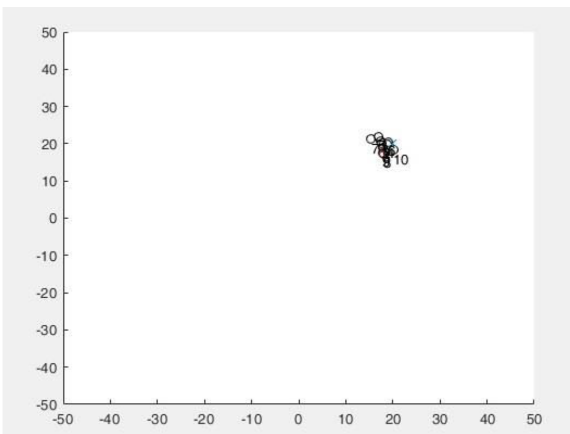
(8)



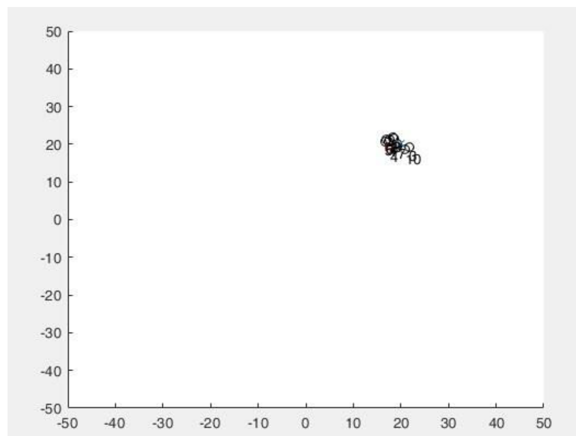
(9)



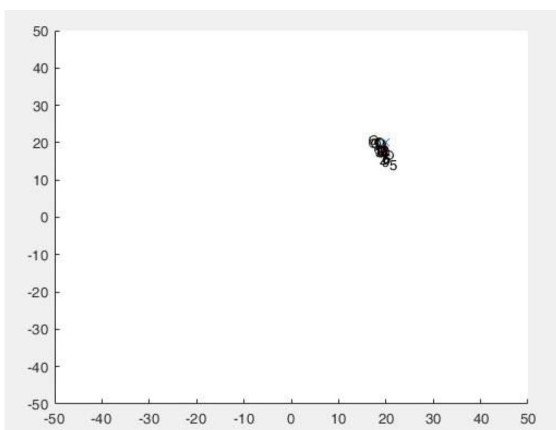
(10)



(11)



(12)



(13)

Fig. 4 Movements of the swarms intended to achieve the best cost

Table 1 shows the achieved best cost as the swarms undergo 25 iterations. Initially, the recorded best cost is 10.9327 which gradually decreases to 2.1745.

Table 1 Achieved Best Cost as swarms undergo execution

Iterations	Best Cost	Cluster Head	Iterations	Best Cost	Cluster Head
1	10.9327	9	14	4.0567	1
2	10.3377	6	15	3.6631	9
3	9.4022	7	16	3.6631	9
4	8.5739	10	17	3.624	5
5	7.7038	5	18	3.624	5
6	7.1531	6	19	2.9879	8
7	5.719	7	20	2.9879	8
8	4.7385	4	21	2.8696	4
9	4.4567	3	22	2.8696	4
10	4.4567	3	23	2.608	6
11	4.3203	2	24	2.608	6
12	4.0567	1	25	2.1745	2
13	4.0567	1			

Fig. 5 shows the graphical representation of the best cost through 25 executed iterations. The graph shows the gradual downfall in context with the recorded best cost.

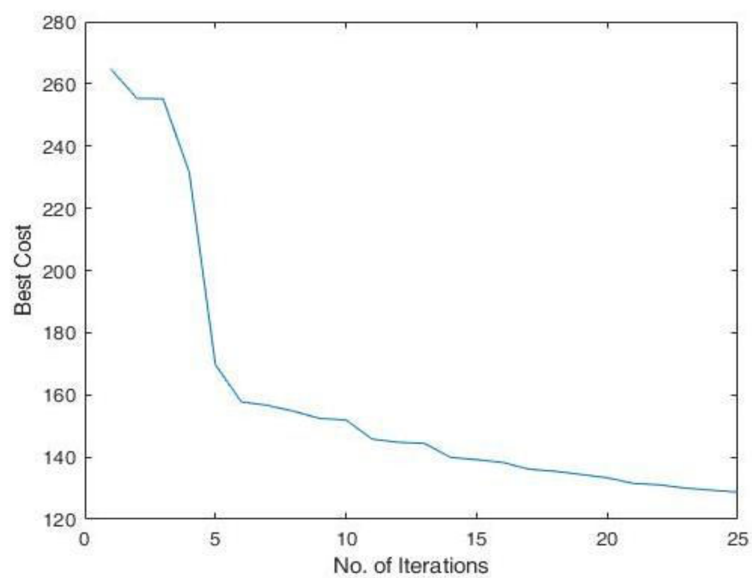


Fig. 5 Graphical representation of the best cost through 25 executed iterations

ABC

The different parameters considered with assigned values to execute the working of PSO are given below:

- Number of Decision Variables (*nVar*) = 5
- Decision Variables Matrix Size (*VarSize*) = [1 nVar]
- Decision Variables Lower Bound (*VarMin*) = -10

- Decision Variables Upper Bound (*VarMax*) = 10
- Maximum Number of Iterations (*MaxIt*) = 25
- Population Size (*nPop*) = 10
- Number of Onlooker Bees (*nOnlooker*) = *nPop*
- Abandonment Limit Parameter (*Trial Limit*)(*L*) = round(0.6*nVar*nPop)
- Acceleration Coefficient Upper Bound (*a*) = 1

Table 2 shows the achieved best cost as the execution proceeds through the 25 iterations. Initially, the recorded best cost is 32.8435 which gradually decreases to .14177.

Table 2 Achieved Best Cost as the execution proceeds through 200 iterations

Iterations	Best Cost		Iterations	Best Cost
1	32.8435		14	1.0048
2	27.1827		15	1.0048
3	27.1827		16	0.98058
4	27.1827		17	0.83622
5	27.1827		18	0.36567
6	25.3088		19	0.36567
7	13.2086		20	0.14177
8	10.2024		21	0.14177
9	10.2024		22	0.14177
10	9.31		23	0.14177
11	2.1137		24	0.14177
12	2.1137		25	0.14177
13	1.0048			

Fig. 6 shows the obtained results in Table II in graphical form. The graph shows the gradual downfall in context with the recorded best cost.

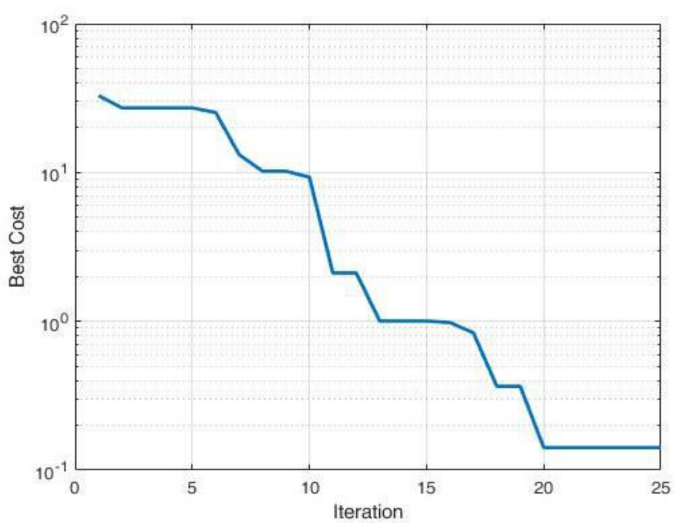


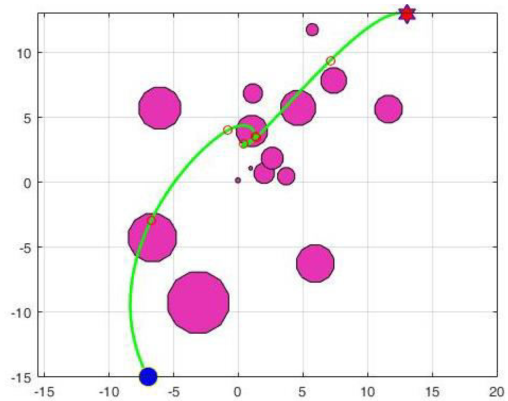
Fig. 6 Graphical representation of the best cost through 25 executed iterations

Hybrid PSOABC Algorithm

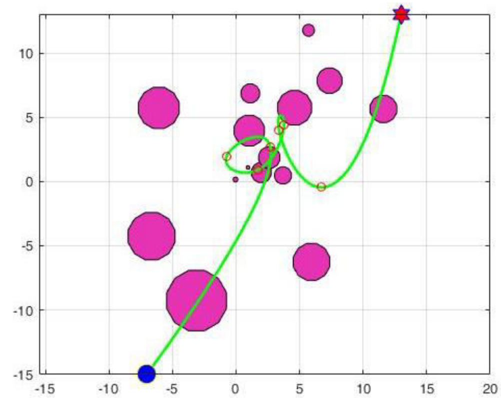
The different parameters considered with assigned values to execute the working of Hybrid PSOABC are given below:

- Number of handle points (*model.n*) = 6
- Number of Decision Variables (*nVar*)= model.n;
- Size of Decision Variables Matrix (*VarSize*)=[1 nVar]
- Lower Bound of Variables (*VarMin.x*)= model.xmin;
- Upper bound of variables (*VarMax.x)*=model.xmax;
- Lower Bound of Variables (*VarMin.y*) = model.ymin
- Upper bound of variables (*VarMax.y*) = model.ymax
- Maximum Number of Iterations (*MaxIt)* =150
- Population size (*nPop*) = 20
- Inertia Weight (*w*) = 1
- Inertia Weight Damping Ratio (*wdamp*) = 0.98
- Personal Learning Coefficient (*c1*) = 1.5
- Global Learning Coefficient (*c2*) = 1.5
- Number of Onlooker Bees (*nOnlooker*) = nPop
- Abandonment Limit Parameter (Trial Limit)(*L)* = round(0.6*nVar*nPop);
- Constant (*a*) = 1
- Constant (*alpha*) = 0.1
- Maximum velocity (*VelMax.x*) = alpha*(VarMax.x-VarMin.x)
- Minimum velocity (*VelMin.x*) = -VelMax.x
- Maximum velocity (*VelMax.y*) = alpha*(VarMax.y-VarMin.y)
- Minimum velocity (*VelMin.y*) = -VelMax.y

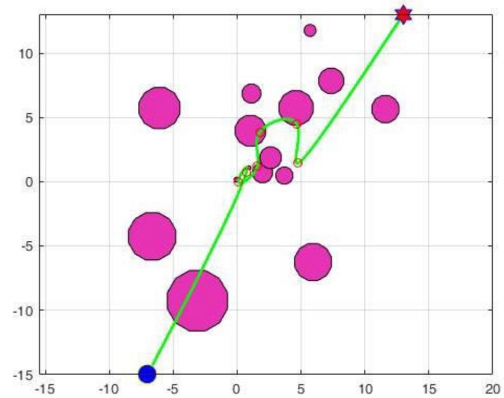
Fig. 7 shows the execution process to find the best route for reaching from source to destination via multiple iterations and shows the search process at different stages for finding the best path. The blue-colored circle is the starting point and the red-colored star is the endpoint. The pink colored circles denote the obstacles between the source and the destination. After undergoing execution for 150 iterations, the best path from source to destination is tracked avoiding the obstacles to the maximum and achieving the optimum fitness value. A higher fitness value indicates a better design. The fitness function has been defined using the best cost function value. Boundary violations have been considered for each iteration and the violation has been mentioned against the iterations where it occurred. The maximum fitness value achieved after the execution of 150 iterations is 1454.2248.


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

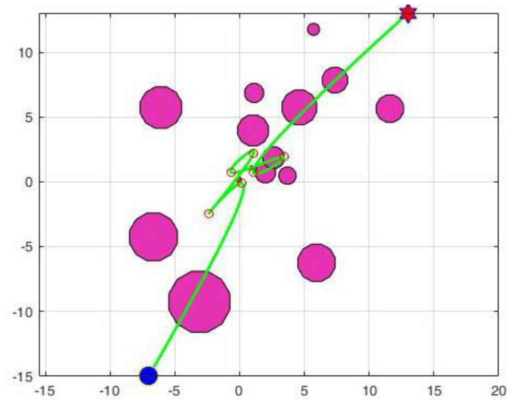
(1)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

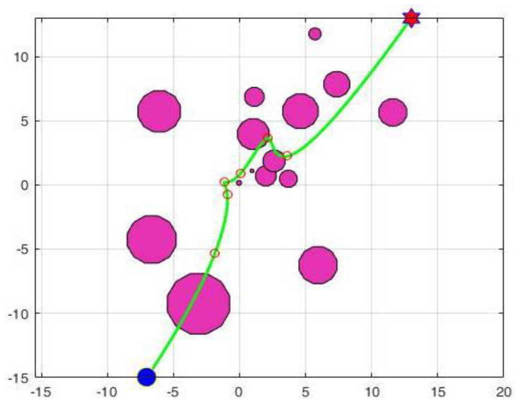
(2)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

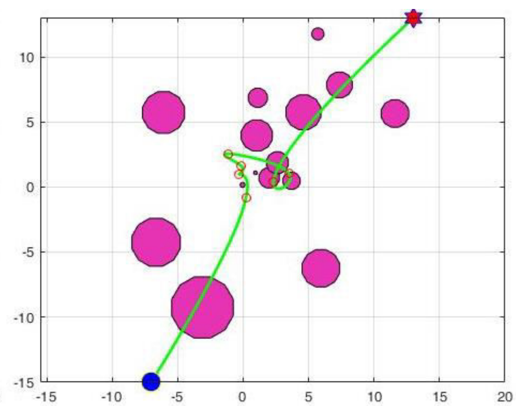
(3)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

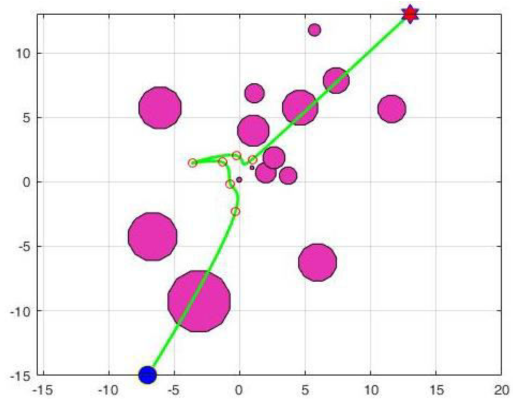
(4)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

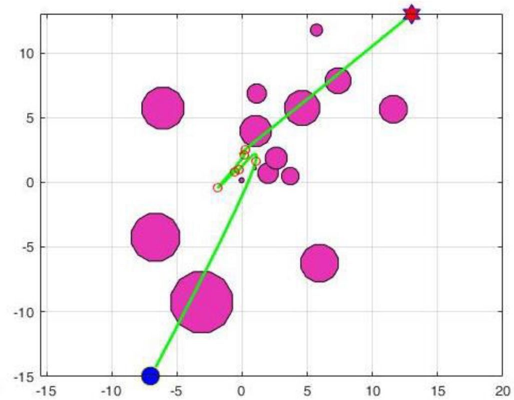
(5)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

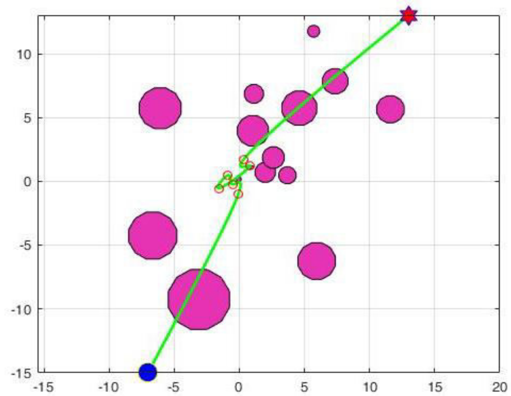
(6)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

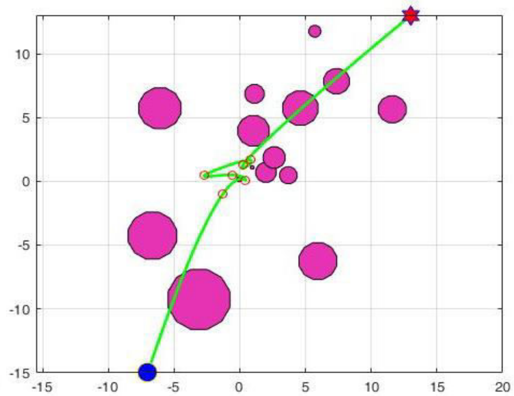
(7)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

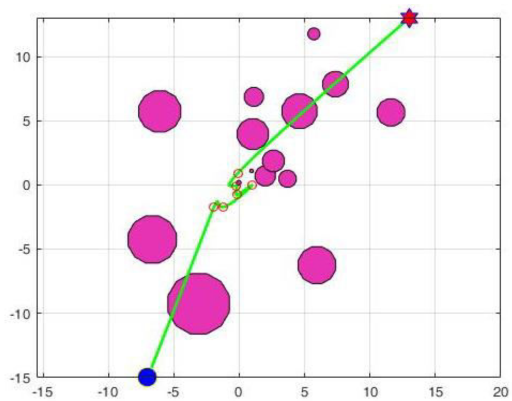
(8)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

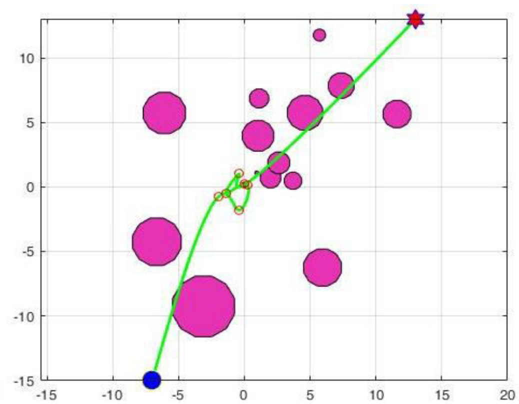
(9)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

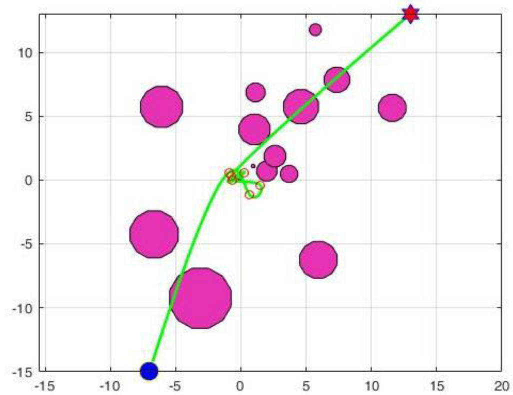
(10)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

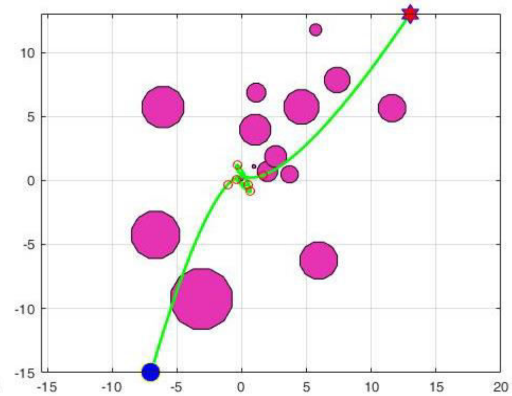
(11)


10
5
0
-5
-10
-15
-15 -10 -5 0 5 10 15 20

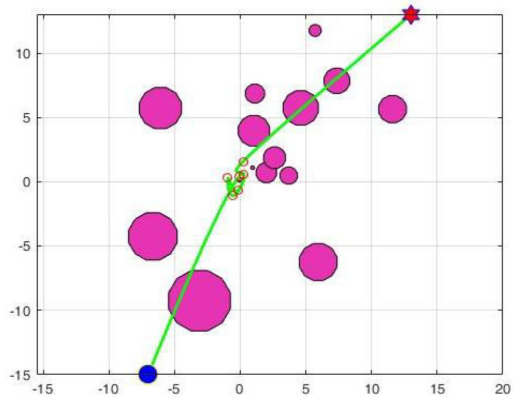
(12)



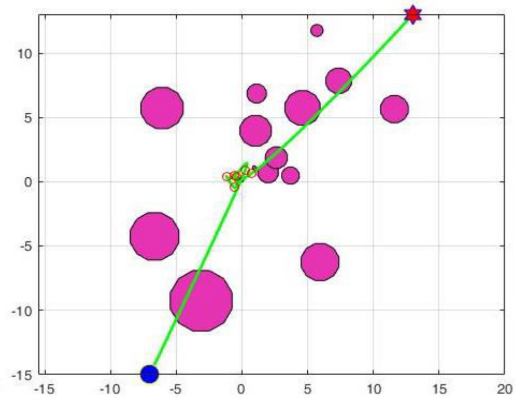
(13)



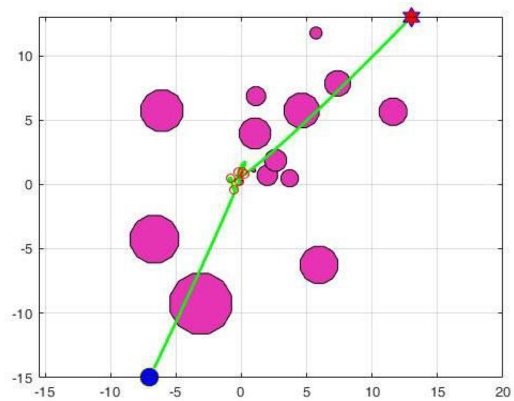
(14)



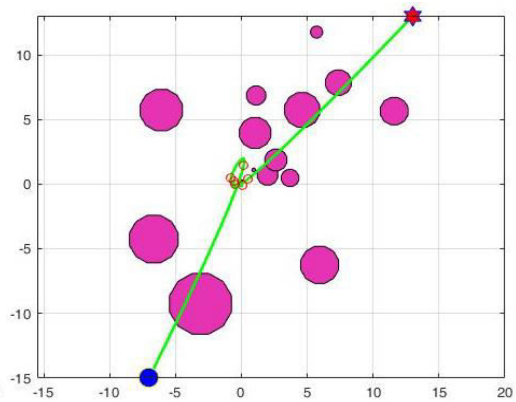
(15)



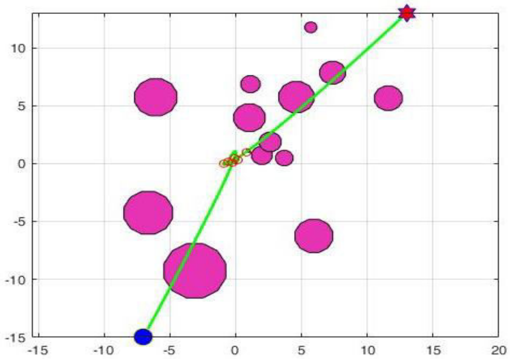
(16)



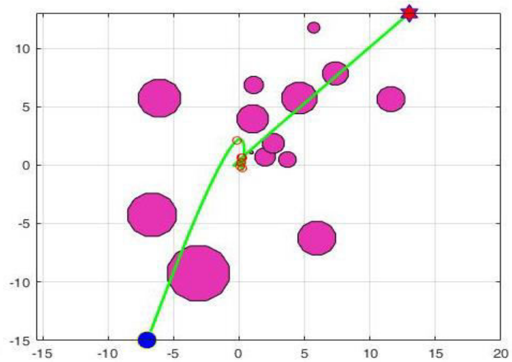
(17)



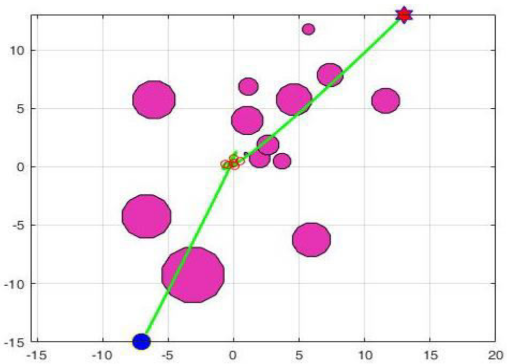
(18)



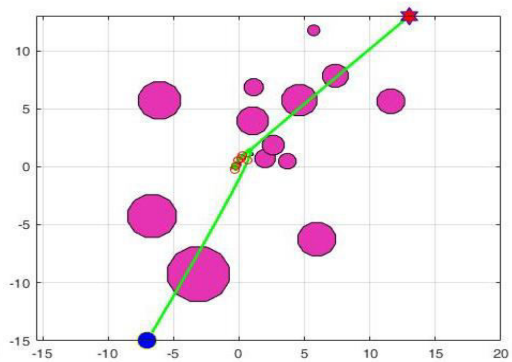
(19)



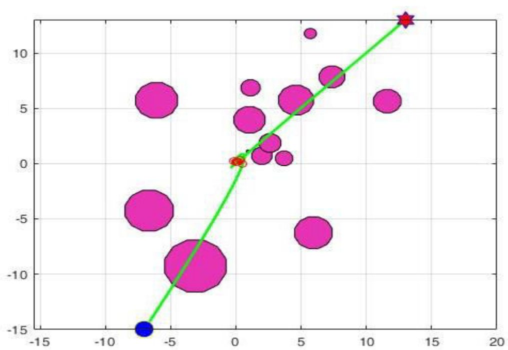
(20)



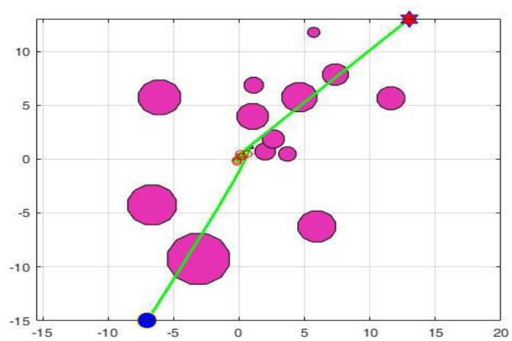
(21)



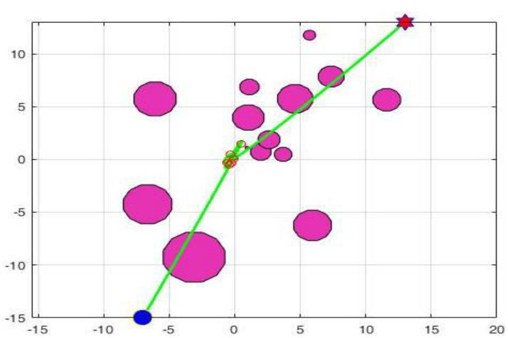
(22)



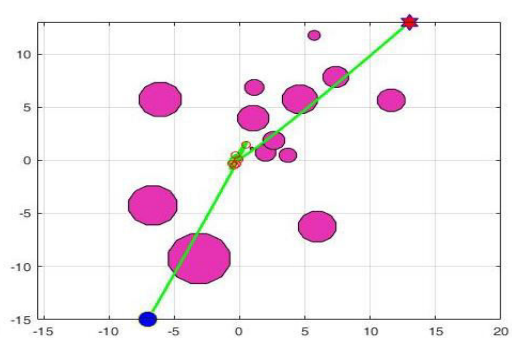
(23)



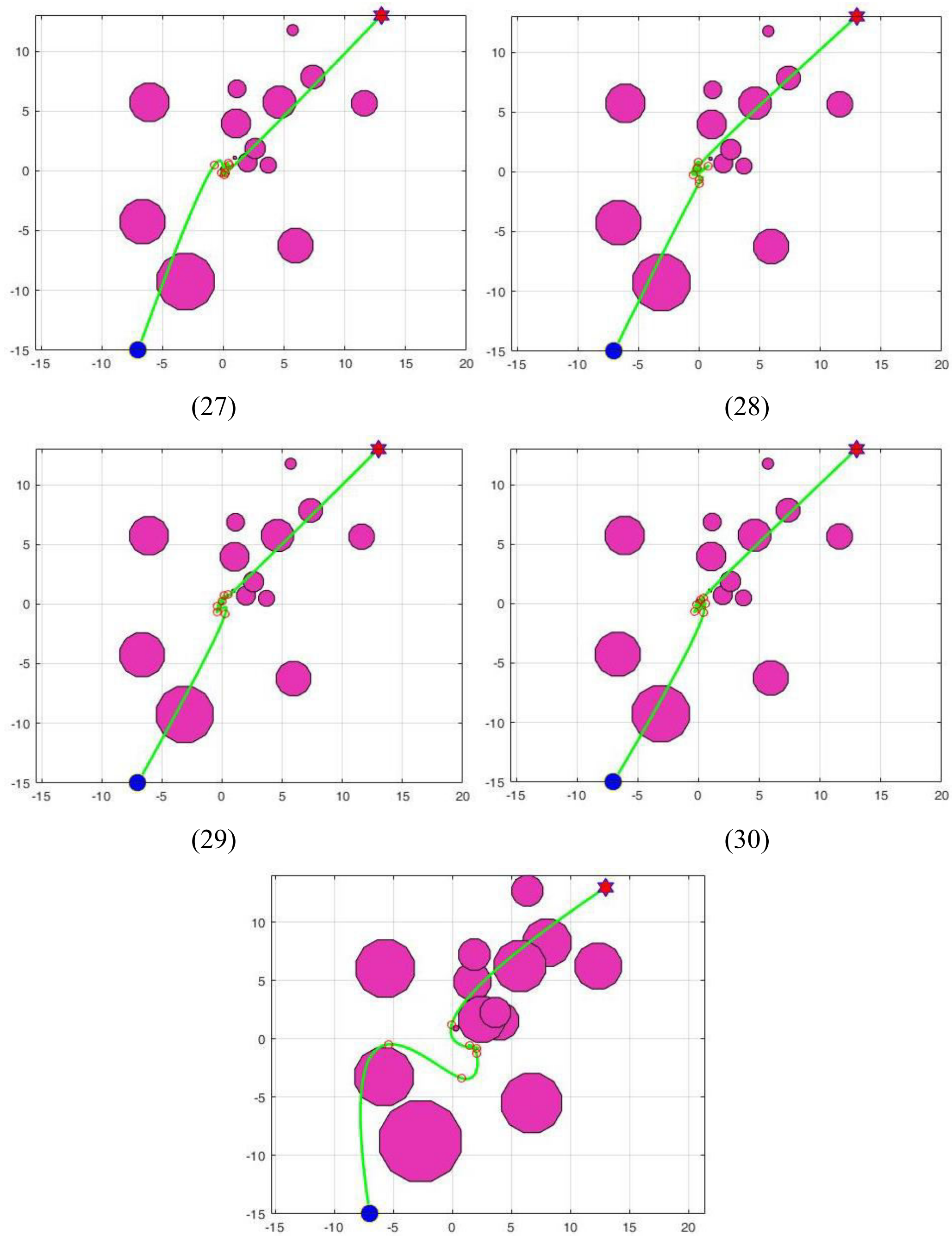
(24)



(25)



(26)



(27) (28)

(29) (30)

(31)

Fig. 7 Execution process to find the best route for reaching from source to destination via multiple iterations

Table 3 depicts the obtained fitness value and the violation as the execution proceeds through the iterations.

Table 3. Fitness value at different iterations

Iteration	Fitness Value	Violation		Iteration	Fitness Value	Violation
1	1306.4065	NULL		136	588.705	4.4327e-07
2	1229.5731	NULL		137	1462.5667	NULL
3	1325.1883	NULL		138	903.3974	NULL
4	1078.5657	NULL		139	904.1474	NULL
5	872.9937	NULL		140	859.7244	NULL
6	868.5799	NULL		141	525.7235	NULL
7	827.992	NULL		142	1462.5655	NULL
8	1205.3726	.0032082		143	904.1451	NULL
9	805.0519	.00054399		144	1228.0083	NULL
10	710.8983	.0046245		145	313.3239	1.3431e-08
11	905.8991	NULL		146	313.3238	NULL
12	2132.9184	.0017662		147	822.2056	NULL
13	1351.0967	.00063469		148	894.1508	NULL
14	1411.4921	8.0312e-05		149	606.7599	NULL
15	1311.2147	NULL		150	1454.2248	NULL
CONTINUE						

V. CONCLUSION AND FUTURE WORK

The algorithms falling under the category of Swarm Intelligence have nearly the same idea and a working concept. All of them are based on population size and the particles are scattered in the solution space. These participating particles have initial locations and as the progress of the algorithm, these particles approach an optimum solution using their swarm intelligence. The research paper discussed and implemented the working of PSO. The best cost achieved gradually decreased from 10.9327 to 2.1745. Similarly, the research methodology based on the working principle of ABC has been developed and implemented to obtain the value of the best cost through 25 iterations. The best cost achieved gradually decreased from 32.8435 to 0.14177. The gradual decrease witnessed in both the algorithms (PSO and ABC) justifies the effective implementation of the adopted methodologies. The Hybrid PSOABC algorithm has been constructed based on PSO and ABC. On execution of the Hybrid PSOABC, it has been found that the fitness value gradually decreases from the first iteration to the last iteration. The decrease in the value of fitness indicates the proper working of the proposed Hybrid PSOABC.

In the future, the number of performance evaluation parameters could be increased. Although improvements have been witnessed in the optimization algorithms using swarm intelligence and meta-heuristics algorithms, a unique algorithm capable of handling all optimization problems effectively and with the best parameters readings is to come up. The development process of hybrid algorithms will continue until the best combination of algorithms would be found.

REFERENCES

Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing – A hierarchical taxonomical classification. *Journal of Cloud Computing*, *8*(1). https://doi.org/10.1186/s13677-019-0146-7

Ahmad, M. O., & Khan, R. Z. (2017). A survey on load balancing algorithms in cloud computing. *International Journal of Autonomic Computing*, *2*(4), 366. https://doi.org/10.1504/ijac.2017.10010817

Arulkumar, V., & Bhalaji, N. (2021). Performance analysis of nature inspired load balancing algorithm in cloud environment. *Journal of Ambient Intelligence and Humanized Computing*, *12*(3), 3735–3742. https://doi.org/10.1007/s12652-019-01655-x

Batista, G. V., Scarpin, C. T., Pécora, J. E., & Ruiz, A. (2019). A New Ant Colony Optimization Algorithm to Solve the Periodic Capacitated Arc Routing Problem with Continuous Moves. *Mathematical Problems in Engineering*, *2019*. https://doi.org/10.1155/2019/3201656

Farid, M., Latip, R., Hussin, M., & Hamid, N. A. W. A. (2020). A survey on QoS requirements based on particle swarm optimization scheduling techniques for workflow scheduling in cloud computing. *Symmetry*, *12*(4). https://doi.org/10.3390/SYM12040551

Jairam Naik, K. (2020). A Dynamic ACO-Based Elastic Load Balancer for Cloud Computing (D-ACOELB). In K. S. Raju, R. Senkerik, S. P. Lanka, & V. Rajagopal (Eds.), *Data Engineering and Communication Technology* (pp. 11–20). Springer Singapore.

Janakiraman, E., & Ramya, V. (2019). *Job Scheduling and Load Balancing In Cloud Computing Using Hacobee*. 64–68.

Jeyafzam, F., Vaziri, B., Suraki, M. Y., Hosseinabadi, A. A. R., & Slowik, A. (2021). Improvement of grey wolf optimizer with adaptive middle filter to adjust support vector machine parameters to predict diabetes complications. *Neural Computing and Applications*, *33*(22), 15205–15228. https://doi.org/10.1007/s00521-021-06143-y

Siddiqui, S., Darbari, M., & Yagyasen, D. (2019). Evaluating load balancing algorithms for performance optimization in cloud computing. *International Journal of Innovative Technology and Exploring Engineering*, *8*(12), 2217–2221. https://doi.org/10.35940/ijitee.L2978.1081219

Tong, Z., Chen, H., Deng, X., Li, K., & Li, K. (2020). A scheduling scheme in the cloud computing environment using deep Q-learning. *Information Sciences*, *512*, 1170–1191. https://doi.org/10.1016/j.ins.2019.10.035

Ullah, A., Nawi, N. M., Uddin, J., Baseer, S., & Rashed, A. H. (2019). Artificial bee colony algorithm used for load balancing in cloud computing: Review. *IAES International Journal of Artificial Intelligence*, *8*(2), 156–167. https://doi.org/10.11591/ijai.v8.i2.pp156-167

Xiao, S., Wang, W., Wang, H., Tan, D., Wang, Y., Yu, X., & Wu, R. (2019). An improved artificial bee colony algorithm based on elite strategy and dimension learning. *Mathematics*, *7*(3), 1–17. https://doi.org/10.3390/math7030289

Yiqiu, F., Xia, X., & Junwei, G. (2019). Cloud Computing Task Scheduling Algorithm Based On Improved Genetic Algorithm. *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, *Itnec*, 852–856. https://doi.org/10.1109/ITNEC.2019.8728996

Yuce, B., Packianather, M. S., Mastrocinque, E., Pham, D. T., & Lambiase, A. (2013). Honey bees inspired optimization method: The bees algorithm. *Insects*, *4*(4), 646–662. https://doi.org/10.3390/insects4040646

Zhang, C., Luo, P., Zhao, Y., & Ren, J. (2019). An efficient round robin task scheduling algorithm based on a dynamic quantum time. *International Journal of Circuits, Systems and Signal Processing*, *13*, 197–204.

Zhang, X., & Hu, Y. (2019). Multiconstrained routing based on artificial bee colony algorithm and dynamic fireworks algorithm. *Journal of Physics: Conference Series*, *1237*(2). https://doi.org/10.1088/1742-6596/1237/2/022058