wjert, 2019, Vol. 5, Issue 6, 272-289. *Original Article* ISSN 2454-695X

World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 5.924

A COMPARATIVE STUDY OF PARTICLE SWARM OPTIMIZATION AND GRAVITATIONAL SEARCH ALGORITHM IN POULTRY HOUSE TEMPERATURE CONTROL SYSTEM

B. O. Ola[1], O. O. Awodoye[2*] and J. P. Oguntoye[2*]

[1]Department of Computer Engineering, Osun State Polytechnic Iree, Osun State, Nigeria.
[2]Department of Computer Science and Engineering, Faculty of Engineering & Technology, Ladoke Akintola University of Technology (LAUTECH), Nigeria.

Article Received on 06/10/2019 **Article Revised on 27/10/2019** **Article Accepted on 17/11/2019**

***Corresponding Author**
O. O. Awodoye
J. P. Oguntoye
Department of Computer Science and Engineering, Faculty of Engineering & Technology, Ladoke Akintola University of Technology (LAUTECH), Nigeria.

ABSTRACT

Low or unexpected rise in temperature is a major factor affect the effectiveness and productivity of broiler chickens. Maintaining and keeping the temperature at normal level is essential to reducing the mortality rate and increase the productivity of the poultry. Some Nature Inspired Algorithms (NIAs) which have proven to be efficient have been adopted to regulate the temperature of the poultry house. However, various studies have shown that there is no algorithm that can achieve the best solution for all optimization problems, and that some algorithms give a better solution for some problems than the others. Therefore, in this study, a comparative analysis of the Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA) in Poultry House Temperature Control System. The experiment results show that both PSO and GSA were able to regulated the poultry house efficiently. However, PSO proved to be more efficient than GSA in terms of cost and computational time. The PSO is able to find better solutions and converges faster compared to the Gravitational Search Algorithm. It is therefore, recommended that PSO should be adopted instead of GSA in poultry house temperature regulation systems.

KEYWORDS: Gravitational Search Algorithm, Nature Inspired Algorithms, Particle Swarm Optimization, Poultry House Temperature, Swarm Intelligence algorithms, Thermal Regulation.

1.0 INTRODUCTION

Poultry constitutes one of the livestock productions that contribute significantly to human source of food (Tilakasiri *et al*., 1988). Poultry farming involves raising of domesticated birds like chickens, ducks, turkeys and geese for the aim of farming meat or eggs for food. Several billions chickens are raised annually as a supply of food, for either meat or eggs (Sinduja *et. al.* 2016).

A comprehensive analysis of studies has shown that comfort satisfaction of poultry birds can be enhanced by dynamically observing various factors like temperature, relative humidity, and solar radiation in the poultry house (William, 1995). Although, it good to control all the above factors, economic considerations have suggested the control of the most important single factor, which is temperature. Adesiji *et. al.* (2013) stated that Ambient temperatures significantly influence the survivability and performance of the poultry production. Poultry flocks are particularly vulnerable to climate change because there is a range of thermal conditions within which animals are able to maintain a moderately steady body temperature in their behavioural and physiological activities. Hence, birds can only tolerate narrow temperature ranges to sustain the peak of their production for human consumption and any unpredictable climatic changes will therefore trigger a series of adjustment and readjustments by livestock and poultry birds in the struggle for survival which may have negative consequence on the viability of poultry production (Adesiji *et. al.*, 2013).

Consequently, there is a need to regulate the temperature of the poultry house at every point in time to ensure optimum poultry production. Several techniques have been adopted to achieve an optimum regulation of the poultry house. The traditional methods despite being effective are time consuming, tedious and require a continuous monitoring (Czarick & Michael, 1994). A good knowledge based system that can regulated the temperature requirement of the poultry system will go a long way to overcome the shortcomings of the existing techniques.

Ola, Oguntoye and Awodoye (2017) evaluated the performance of Particle Swarm Optimization (PSO) on poultry house temperature control system. The PSO technique was found to be computationally efficient. In this study, a comparative analysis of the performance of Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA) in poultry house temperature control system will be carried out.

Particle Swarm Optimization (PSO) is a computational search and optimization technique that have been empirically shown to achieve well on several optimization problems. It is extensively used to find the worldwide optimum explanation and solution in a complex search space (Kumar, Singh & Patro, 2016). PSO is a population-based optimization technique inspired by the behaviour of schools of fish, herds of animals or flocks of birds (Eberhart & Kennedy 1995).

Gravitational Search Algorithm (GSA) is a stochastic population-based metaheuristic inspired by the interaction of masses via Newtonian gravity law. It is an optimization algorithm that is based on the law of gravity and mass interactions. In this algorithm, the searcher agents are a collection of masses, and their interactions are based on the Newtonian laws of gravity and motion **(**Rashedi *et. al.*, 2010).

Both PSO and GSA can be regarded as a Nature Inspired Algorithms (NIAs) as well as Swarm intelligence algorithms. NIAs are inspired by nature and used to deal with difficult real-world engineering problems **(**Rashedi *et. al.*, 2009). Swarm intelligence algorithms are inspired by any type of collective behaviours of individuals in nature (Bansal *et. al.*, 2014). Over the years, there has been an increasing concentration in algorithms inspired by the observation of natural phenomena **(**Rashedi *et. al.*, 2010). Several researches have validated that these algorithms are good alternatives as tools to solve difficult computational problems. Numerous heuristic methodologies such as Artificial Immune System (AIS), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Simulated Annealing (SA), Gravitational search algorithm (GSA), and Particle Swarm Optimization (PSO) have been adopted by researches in various applications. Nevertheless, various studies have shown that there is no algorithm that can achieve the best solution for all optimization problems, and that some algorithms give a better solution for some problems than the others (Engelbrecht, 2005) (Cheng *et. al.*, 2007). Hence, the need for comparative analysis between the performance of Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA) in poultry house temperature control system.

2.0 Literature Review

2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization Algorithm (PSO) is one of swarm intelligence optimization algorithms. It belongs to class of random searching algorithms (Adel & Songfeng, 2016). The main idea behind PSO is originated from the sharing and updating of information among bird (particle) individuals in the process of searching food. Each individual bird can benefit from discovery and flight experience of the others. In PSO algorithms, the particle swarm is initialized randomly in searching space and each particle has initial speed and position. So the searching quality and the speed have randomness. The path of particle is updated through individual best position and the path of swarm is updated via global best location, which is found by the entire population. This makes particles move to the optimal solution (Zhang *et. al.*, 2013). Each particle updates its velocity and position according to equation (1) and (2) respectively (Attiya & Zhang, 2017):

$$\vec{v}_{ij}(t+1) = \omega\vec{v}_{ij}(t) + c_1r_1\left(P_b - \vec{x}_{ij}(t)\right) + c_2r_2\left(G_b - \vec{x}_{ij}(t)\right) \qquad (1)$$

$$\vec{x}_{ij}(t+1) = \vec{x}_{ij}(t) + \vec{v}_{ij}(t+1) \qquad (2)$$

Where $\vec{v}_{ij}(t+1)$, velocities of particle i at iterations j, $\vec{x}_{ij}(t+1)$, positions of particle i^{th} at iterations j^{th}. ω is inertia weight to be employed to control the impact of the previous history of velocities. t denotes the iteration number, c_1 is the cognition learning factor, c_2 is the social learning factor, r_1 and r_2 are random numbers uniformly distributed in [0, 1].

The Stepwise procedure of the PSO algorithm are presented below (Alam *et. al.*, 2015):

1. Set parameter ω_{min}, ω_{max}, c_1 and c_2 of PSO
2. Initialize population of particles having positions x_j and velocities v_j
3. Set iteration k = 1
4. Calculate fitness of particles $F_{ij}(t) = f(\vec{x}_{ij}(t))$ and find the index of the best particle *b*
5. Select $Pbest_{ij}(t) = \vec{x}_{ij}(t)$ and $Gbest_j(t) = x_{bj}(t)$
6. $\omega = \omega_{max} - k \times (\omega_{max} - \omega_{min})/Max\_no$
7. Update velocity and position of particles

$$\vec{v}_{ij}(t+1) = \omega\vec{v}_{ij}(t) + c_1r_1\left(P_b - \vec{x}_{ij}(t)\right) + c_2r_2\left(G_b - \vec{x}_{ij}(t)\right)$$

$$\vec{x}_{ij}(t+1) = \vec{x}_{ij}(t) + \vec{v}_{ij}(t+1)$$

8. Evaluate fitness $F_{ij}(t+1) = f(\vec{x}_{ij}(t+1))$ and find the index of the best particle b_1

9. Update $Pbest$ of population

If $F_{ij}(t+1) < F_{ij}(t)$ then $Pbest_{ij}(t+1) = \vec{x}_{ij}(t+1)$ else

$Pbest_{ij}(t+1) = Pbest_{ij}(t)$

10. Update $Gbest$ of population

If $F_{bj}(t+1) < F_{bj}(t)$ then $Gbest_j(t+1) = Pbest_{bj}(t+1)$ and set $b = b_1$ else

$Gbest_{bj}(t+1) = Gbest_j(t)$

11. If $k < Max\_no$ then $k = k+1$ and goto step 6 else goto step 12

12. Output optimum solution as $Gbest_j(t)$.

2.2 Gravitational Search Algorithm (GSA)

Gravitational search algorithm (GSA) is a simple well known meta-heuristic search algorithm based on the law of gravity and the law of motion **(**Rashedi *et. al.,* 2009). It is a swarm intelligence type algorithm that is inspired by the Newton's physics concept gravitational force and motion of individuals in nature. Like many other nature-inspired algorithms, it needs refinements to maximize its performance in solving various types of problems. In addition to the problem encoding that sometimes can be a challenge, fine tuning its parameters play a significant role balancing the search time versus solution quality (Taisir & Al Qasim, 2013). GSA is an optimization algorithm and provides proper balancing between exploitation and exploration capabilities. So in this algorithm, heavier masses individuals are responsible for exploitation whereas lighter masses individuals are responsible for the exploration of the search area (Aditi *et. al*., 2017). When searching process start lighter masses (individuals are far from the optimum solutions) individuals move with large step size (exploration) and after this when individuals converge to the optimum solutions i.e. higher masses individuals move with comparative small step size (exploitation).

The GSA algorithm is described by the following steps **(**Rashedi *et. al.,* 2009):

Step 1: Agents initialization:

The positions of the N number of agents are initialized randomly.

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n) \;\; for\; i = 1, 2, \ldots, N \qquad (3)$$

x_i^d represents the positions of the i^{th} agent in the d^{th}dimension, while n is the space dimension.

Step 2: Fitness evolution and best fitness computation:

For minimization or maximization problems, the fitness evolution is performed by evaluating the best and worst fitness for all agents at each iteration.

For minimization problems best and worst fitness are:

$$best(t) = min\ fit_j(t) \quad j \in 1, \ldots, N \tag{4}$$

$$worst(t) = max\ fit_j(t) \quad j \in 1, \ldots, N \tag{5}$$

For maximization problems best and worst fitness are:

$$best(t) = max\ fit_j(t) \quad j \in 1, \ldots, N \tag{6}$$

$$worst(t) = min\ fit_j(t) \quad j \in 1, \ldots, N \tag{7}$$

$fit_j(t)$ represents the fitness value of the i^{th} agent at iteration t, $best(t)$ and $worst(t)$ represents the best and worst fitness at iteration *t*.

Step 3: Gravitational constant (*G*) computation:

The gravitational constant G*(t)* is computed using the equation below.

$$G(t) = G_0 e^{(-\alpha t/T)} \tag{8}$$

G_0 and α are initialized at the beginning and will be reduced with time to control the search accuracy. *T* is the total number of iterations.

Step 4: Calculation of the Masses of the agents:

Gravitational and inertia masses for each agent are calculated at iteration *t*. Masses in GSA depend upon the fitness value of agents.

$$M_{aj} = M_{pi} = M_{ii} = M_i \qquad i = 1, 2, \ldots, N \tag{9}$$

$$m_i(t) = \frac{fit_i - worst(t)}{best(t) - worst(t)} \tag{10}$$

$$M_i = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{11}$$

Where M_{ii} and M_{pi} are inertia and passive gravitational masses of i^{th} agent respectively and M_{aj} is active gravitational mass of j^{th} agent. fit_i is the fitness value of i^{th} agent.

Step 5: Calculation of Agent's Accelerations:

The acceleration of agents are calculated using the equation below:

$$a_i^d(t) = F_i^d(t)/M_{ii}(t) \tag{12}$$

$F_i^d(t)$ is the total force acting on i^{th} agent calculated as:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq 1} rand_j F_{ij}^d(t) \tag{13}$$

$Kbest$ is the set of first K agents with the best fitness value and biggest mass. $Kbest$ will reduce in each iteration and at the end only one agent applying force to the other agents.

Force on i^{th} agent by d^{th} agents mass during iteration t is computed using the following equation:

$$F_{ij}^{d}(t) = G(t).\left(\frac{M_{pi}(t)xM_{aj}(t)}{R_{ij}(t)} + \varepsilon\right).\left(x_j^d(t) - x_i^d(t)\right) \qquad (14)$$

$R_{ij}(t)$ is the Euclidian distance between two agents i and j at iteration t . $G(t)$ is the gravitational constant calculated using equation 8 while ε is a small constant.

Step 6: Velocity and positions of agents:

The velocity update equation for agents is defined as

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \qquad (15)$$

$rand$ is random variable in interval [0,1]. $v_i^d(t)$ and $v_i^d(t+1)$ are the velocity of i^{th}individual during the iteration t and $t+1$ respectively. The position update equation for individuals is defined as:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \qquad (16)$$

$x_i^d(t)$ and $x_i^d(t+1)$ are the position of i^{th} individual during the iteration t and $t+1$ respectively. Velocity of individuals is updated during each iteration. Due to changes in the velocity every individual updates its position.

Step 7: Repeat steps 2 to 6

Steps 2 to 6 are repeated until the iterations reach their maximum limit. The best fitness value at the final iteration is computed as the global fitness while the position of the corresponding agent at specified dimensions is computed as the global solution of that particular problem.

2.3 GSA versus PSO

In both GSA and PSO the optimization is achieved by agent's movement in the search space, nevertheless the movement strategy is different. Some significant differences are as follows **(**Rashedi *et. al.*, 2009):

1. In PSO the direction of an agent is calculated using only two best positions, pbesti and gbest. But in GSA, the agent direction is calculated based on the overall force obtained by all other agents.
2. In PSO, updating is performed without considering the quality of the solutions, and the fitness values are not important in the updating procedure while in GSA the force is proportional to the fitness value and so the agents see the search space around themselves in the influence of force.

3. PSO uses a kind of memory for updating the velocity (due to pbesti and gbest). However, GSA is memory-less and only the current position of the agents plays a role in the updating procedure.
4. In PSO, updating is performed without considering the distance between solutions while in GSA the force is reversely proportional to the distance between solutions.
5. Finally, note that the search ideas of these algorithms are different. PSO simulates the social behavior of birds and GSA inspires by a physical phenomenon **(**Rashedi *et. al.*, 2009).

3.0 MATERIAL AND METHODS

In this study, a comparative analysis of GSA and PSO for optimal control of poultry house temperature is presented. Each of the techniques is developed to either activate or deactivate either the air conditioning system or the heater as the case may be at a point in time. Red, Green and Yellow light indicate the state of temperature in the poultry house. The red signal indicate that the temperature is below or above the normal temperature. The green signal indicates that the temperature is normal while the yellow signal indicates warning signal that the temperature is either tending above or below the normal temperature. RT, GT and YT represent the red timer, green timer and yellow timer respectively. "**R**", "**Y**" and "**G**" are the red, yellow and green signals respectively (ola *et. al.* 2017).

The optimization of the initial parameter and random number representing the current temperature of the poultry house at a particular point in time is done with respect to the selected technique i.e. PSO or GSA. The output of the best randomised set of temperature from either PSO or GSA regulates the poultry system temperature. Figure 1 and 2 depict the process of Gravitational search algorithm (GSA) and Particle Swarm Optimization (PSO) respectively.

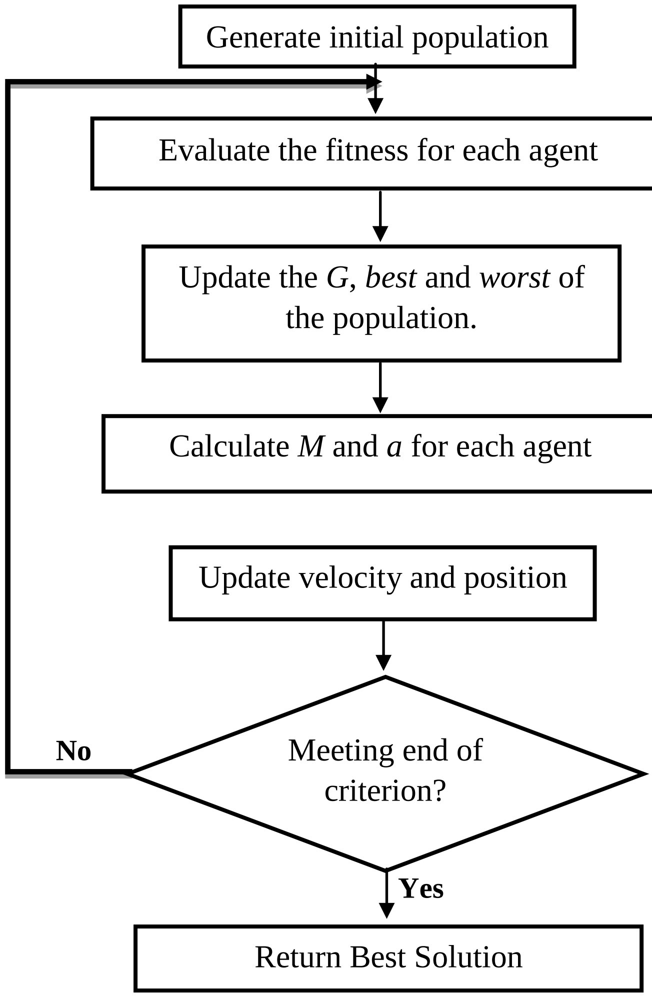

Figure 1: General principle of GSA.

Three variables used to represent levels of temperature include: **Below NT, NT and Above NT** which represent Below Normal Temperature, Normal Temperature and Above Normal Temperature respectively.

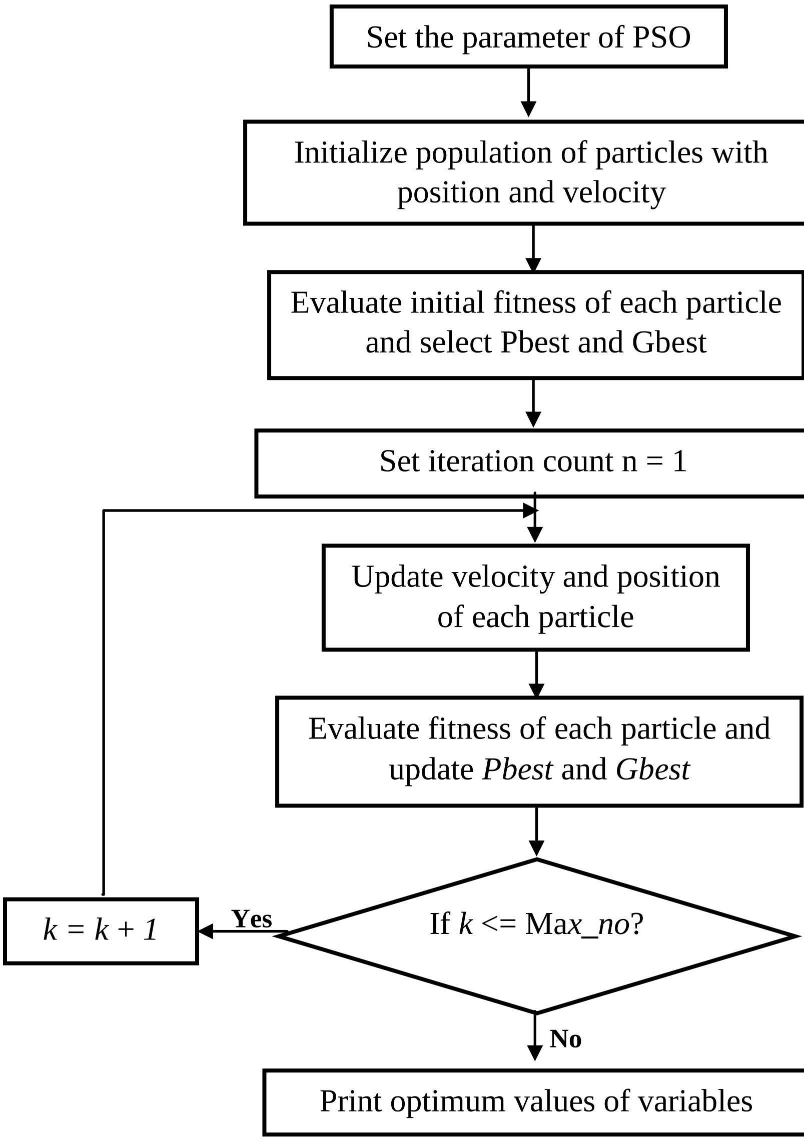

Figure 2: Particle Swarm Optimization Process.

The flow diagram of the program activity is shown in Figure 3. The inputs which includes the current temperature and various parameters needed to control the temperature of a broiler chickens in a poultry farm were initialized. These inputs include: "**TN**" which indicates the total number of cycles and is initialized to be 10. "**C**" is the counter for the total number of cycles and is initialized to be 1. RT, GT and YT represent the red timer, green timer and yellow timer respectively. "**k**" is the counter for the iteration and it is set to 1. "**R**", "**Y**" and "**G**" are the red, yellow and green signals respectively. These parameters were presented as inputs to either GSA or PSO algorithm for optimization. The optimized parameter from either GSA or PSO regulate the thermal requirement of a poultry house.

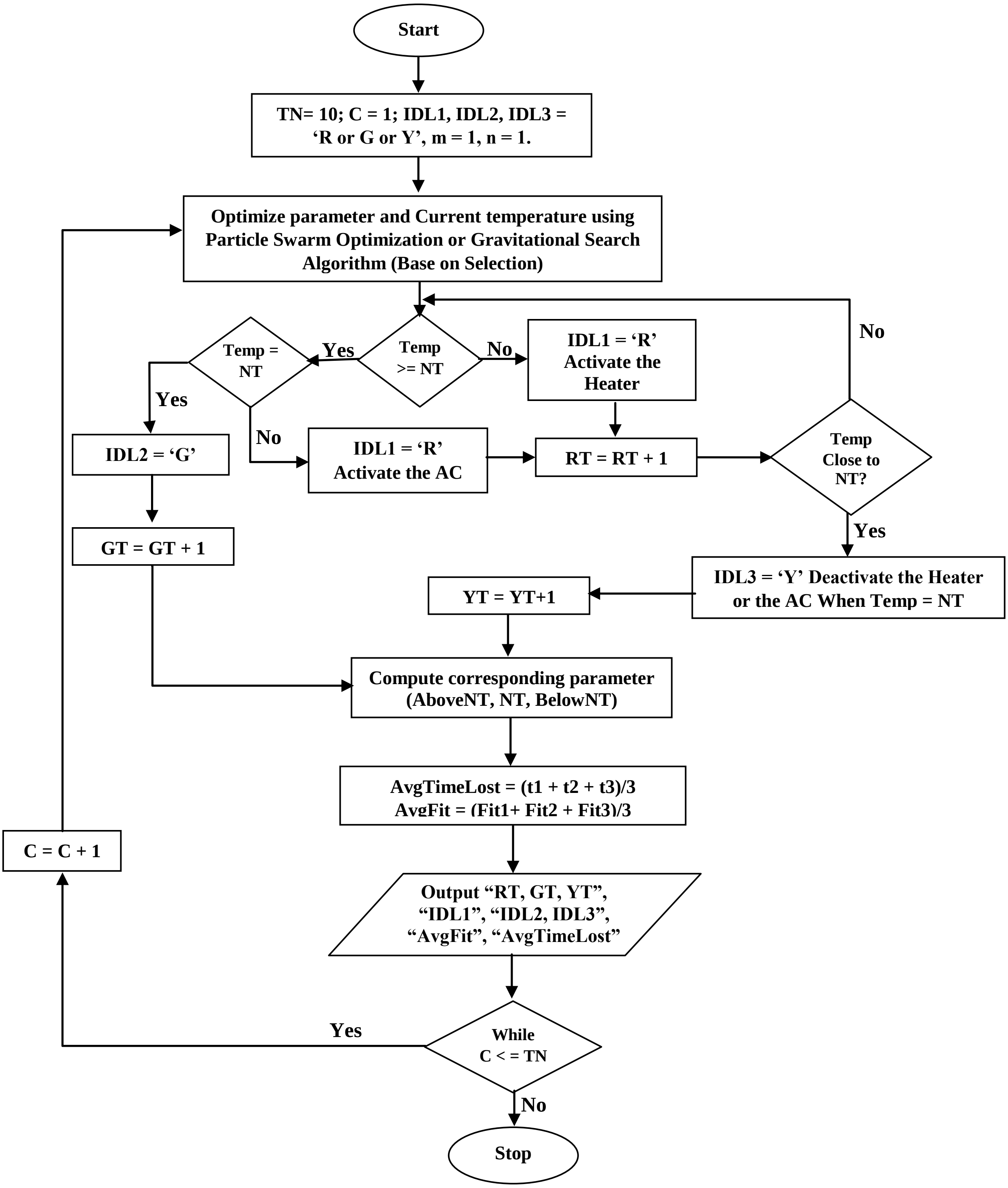

Figure 3: Program Activity of the model.

When the current temperature is greater than the normal temperature (NT) i.e. (Temp = AboveNT) the AC is activated to lower the temperature; the red indicator "**R**" becomes ON and the red timer is increased. Similarly, when the current temperature is less than the normal temperature (NT) i.e. (Temp = BelowNT) the Heater is activated to raise the temperature; the red indicator "**R**" becomes ON and the red timer is increased as well (ola *et. al.* 2017).

At a point when the temperature is very close to the normal temperature as a result of heating or air-conditioning the warning signal is ON i.e. the yellow indicator "**Y**" and the yellow timer is increased. However, when the temperature become normal either the AC or the heater is deactivated. When the temperature is normal i.e. (Temp = NT) the AC and heater will be deactivated while the green indicator "**G**" becomes ON and the green timer is increased.

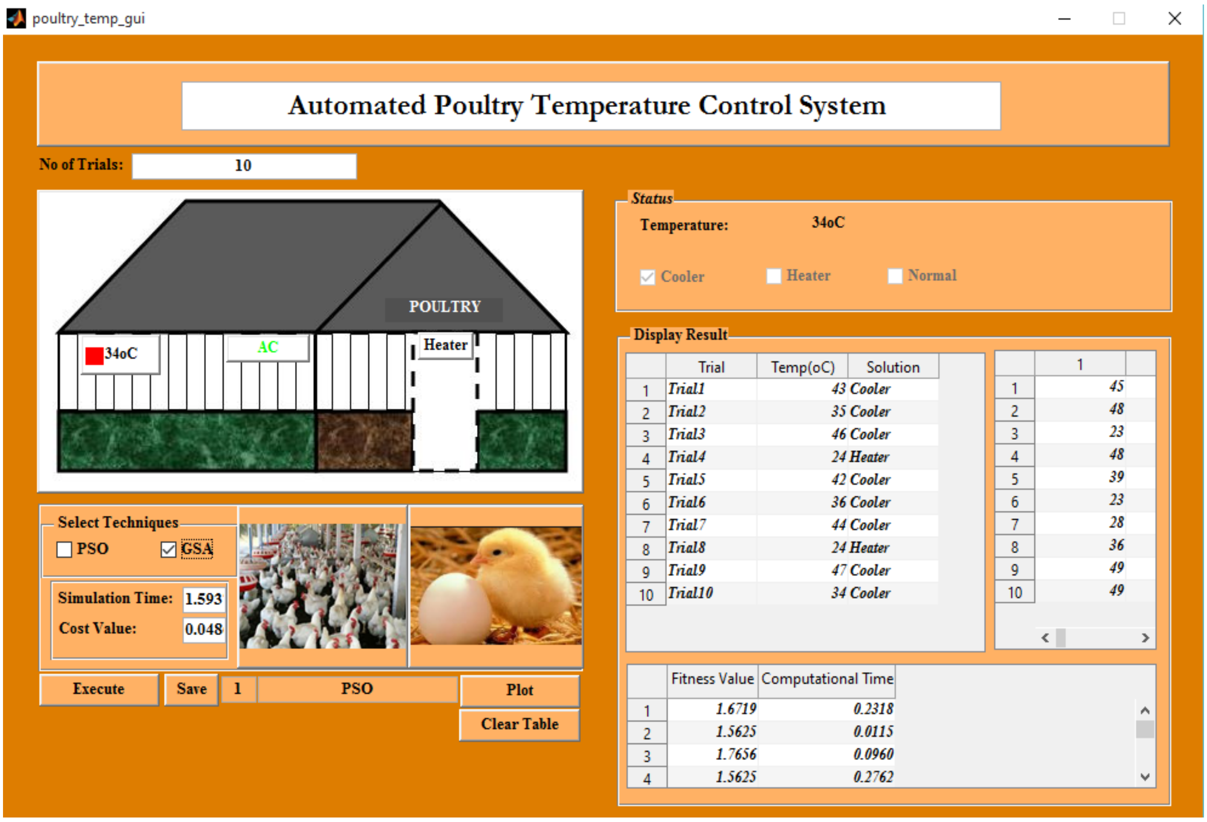

Figure 4: The GUI Application.

The fitness value is Ft1, Ft2 and Ft3 and the time lost in each instance are t1, t2 and t3. If the maximum criteria are met, the model outputs the temperature condition of the poultry system (AboveNT, NT and BelowNT), the average total time in each cases; the average fitness values for each cycle and the average time lost. The fitness values of each cycle are stored with the corresponding average time lost during the temperature regulation. The cycle with the least time lost has the best fitness value. MATLAB R2012a on Windows 10 64-bit operating system, Intel®Pentium® CPU T4500@2.30GHZ Central Processing Unit, 4GB

Random Access Memory and 500GB hard disk drive was used to implement the proposed work. An interactive Graphic User Interface (GUI) application was developed to ensure easy interaction and understanding as shown in figure 4.

4.0 RESULT AND DISCUSSION

The experiment was carried out for each technique for poultry house temperature regulation. The results obtained by the two techniques i.e. GSA and PSO were presented below. Table 1 and 2 shows the results obtained by the application PSO and GSA respectively.

Table 1: PSO Result for Poultry House Temperature Regulation.

Trials	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8	Cycle 9	Cycle 10
	^oC	^oC	^oC	^oC	^oC	^oC	^oC	^oC	^oC	^oC
1	45	20	22	43	47	38	35	20	41	43
2	48	27	31	49	32	44	47	46	20	35
3	23	22	20	39	29	24	35	23	40	46
4	48	27	41	41	28	25	47	23	44	24
5	39	43	25	35	22	27	20	27	23	25
6	23	25	27	33	25	37	38	47	25	36
7	25	36	38	29	35	20	34	33	48	44
8	36	44	45	27	32	48	28	25	49	24
9	49	36	25	39	36	47	36	50	33	47
10	49	50	33	21	48	31	28	28	42	34

The simulated result shows that both the GSA and the PSO technique were able to regulated the temperature of the poultry house by switching on or off either the heater or cooler as the case maybe at every point in time. Also, with respect to the result obtainable from the tables it the observed that the temperature regulation was done mostly by the air-conditioning system. Figure 5 and 6 illustrate the graph of the temperature with respect to 10 regulation attempt for the cycle with the best fitness value and computation time for PSO and GSA respectively. Table 3 present the results of PSO and GSA in terms of fitness value and the computational time. PSO and GSA achieved an average fitness value of 0.090 and 1.075 at 1.61 and 8.22 seconds. Respectively. Therefore, the difference between the PSO and the GSA techniques in terms of both the fitness value and the computational time is 6.61 seconds and 0.98. The fitness value of GSA was constant for all cycles while that of PSO varies but with better cost value. The above result establishes the fact that PSO is more computational efficient in terms of both computation time and fitness value in the regulation of poultry house temperature.

Table 2: GSA Result for Poultry House Temperature Regulation.

Trials	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8	Cycle 9	Cycle 10
	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C
1	44	48	48	37	47	45	45	50	22	47
2	47	30	36	20	34	42	48	49	46	34
3	22	40	25	24	46	45	23	35	30	23
4	42	48	32	28	37	42	48	44	35	38
5	27	42	22	42	22	39	39	24	28	24
6	44	25	48	35	45	33	23	33	26	46
7	32	45	30	26	37	28	28	48	39	50
8	24	25	47	46	27	24	36	44	40	37
9	46	32	35	39	38	43	49	49	41	28
10	44	22	45	32	30	32	49	24	33	48

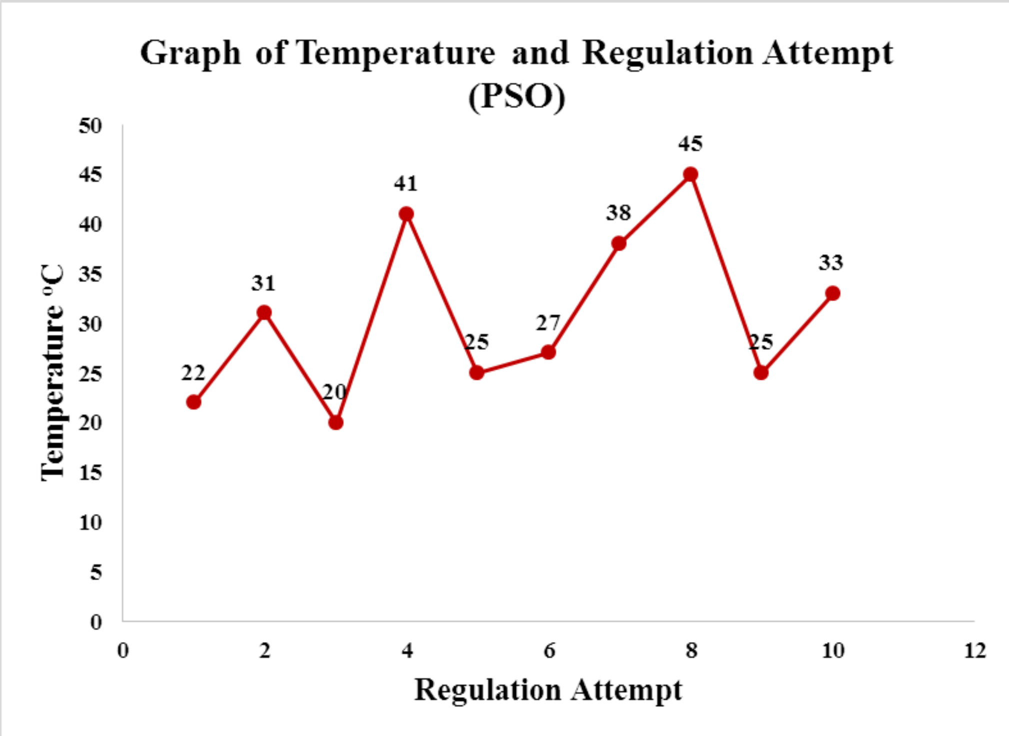

Figure 5: Graph of Temperature against Regulation Attempt for PSO.

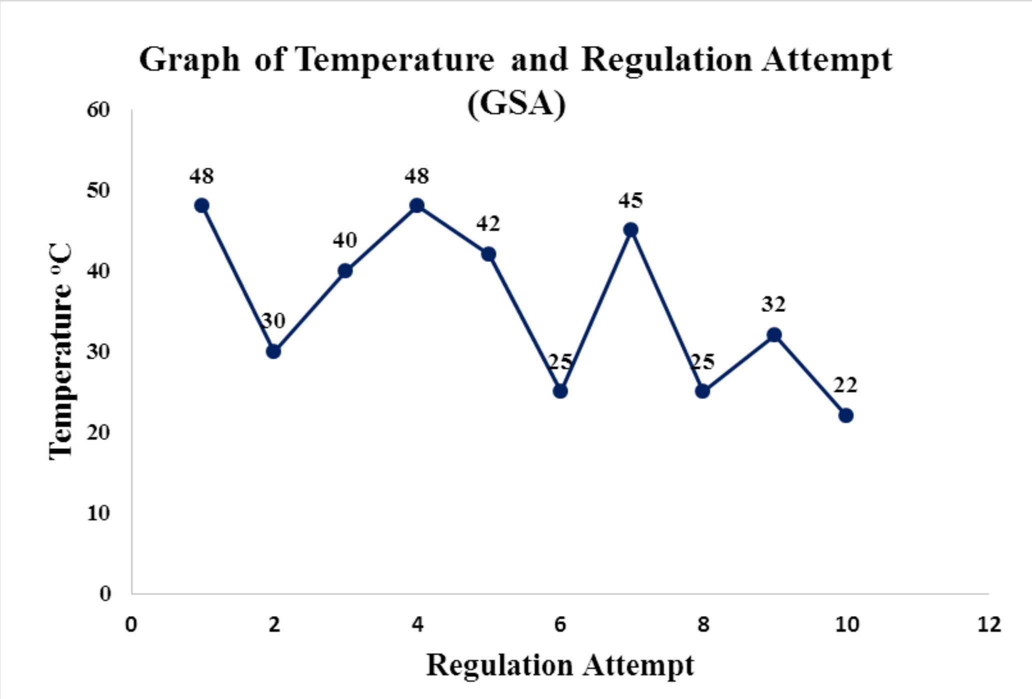

Figure 6: Graph of Temperature against Regulation Attempt for GSA.

Table 3: Simulated PSO and GSA results Showing Fitness Value and Computation Time.

Cycle	**Computation Time (Second)**		**Fitness or Cost Value**	
Method	**PSO**	**GSA**	**PSO**	**GSA**
1	1.67	8.41	0.232	1.075
2	1.56	7.56	0.028	1.075
3	1.52	8.56	0.012	1.075
4	1.56	8.31	0.276	1.075
5	1.77	8.11	0.096	1.075
6	1.56	8.00	0.115	1.075
7	1.59	8.47	0.013	1.075
8	1.61	8.22	0.032	1.075
9	1.67	8.41	0.052	1.075
10	1.59	8.13	0.049	1.075
Average	**1.61**	**8.22**	**0.090**	**1.075**

Figure 7 and 8 illustrate the graph of computation time per cycle and fitness value per cycle for both PSO and GSA respectively. The experimental result shows that PSO is able to find better solutions and converges faster compared to GSA. Consequently, PSO will provide a more reliable poultry house temperature regulation system compared to GSA.

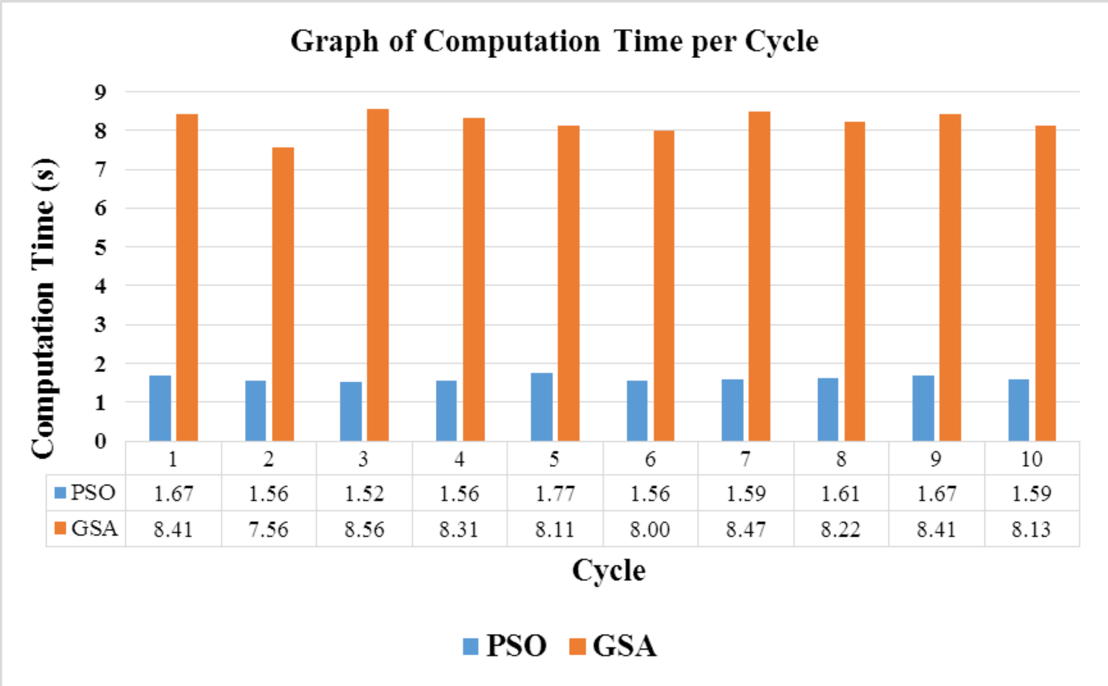

Figure 7: Graph of Computation Time per Cycle.

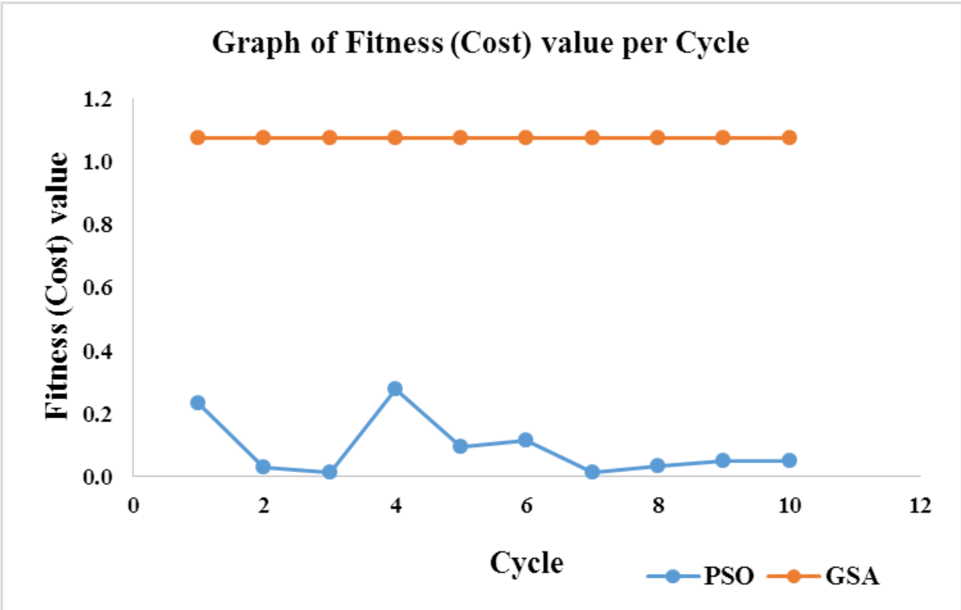

Figure 8: Graph of Fitness Value per Cycle.

CONCLUSION

In this paper, a comparative analysis between the performance of Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA) in poultry house temperature control system was carried out. MATLAB R2012a was used in the design and implementation of the techniques. The experimental result obtained reveals that both PSO and GSA techniques are efficient in the regulation of the temperature of the poultry house for optimum production. However, PSO was found to be more efficient than GSA in terms of cost and computational time. It is therefore, recommended that PSO should be adopted instead of GSA in poultry house temperature regulation systems. Other techniques like Artificial Immune Systems (AIS), Ant Colony Optimization (ACO) can be applied and compared with Particle Swarm Optimization (PSO) to determine their computational efficiency in the regulation of temperature requirement for poultry house.

REFERENCES

1. Adel H. A. & Songfeng L. U. (2016). “A Particle Swarm Optimization Algorithm Based on Uniform Design.” International Journal of Data Mining & Knowledge Management Process (IJDKP) 6(2): 1-6.
2. Adesiji, G. B., Tyabo. I. S., Bolarin, O. Ibrahim, M. & Baba, S. T. (2013). “Effects of Climate Change on Poultry Production in Ondo State, Nigeria.” Ethiopian Journal of Environmental Studies and Management, 6(3): 2-4.
3. Aditi G., Nirmala S., & Harish S. (2017). “Exploitative Gravitational Search Algorithm.” Proceedings of Sixth International Conference on Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing 546, DOI 10.1007/978-981-10-3322-315.
4. Alam M. N., Das B. & Pant V. (2015). “A comparative study of metaheuristic optimization approaches for directional overcurrent relays coordination.” Electric Power Systems Research, 128: 39-52.
5. Attiya I. & Zhang X. (2017). “A Simplified Particle Swarm Optimization for Job Scheduling in Cloud Computing.” International Journal of Computer Applications 163(9): 34-41.
6. Bansal, J. C., Sharma, H., Arya, K.V., Deep, K., Pant, M. (2014). “Self-Adaptive Artificial Bee Colony. Optimization 63(10): 1513–1532.
7. Cheng Y. M., Li L. et al (2007). “Performance studies on six heuristic global optimization methods in the location of critical slip surface.” Computers and Geotechnics. 34(6): 462-484.
8. Czarick M. & Michael L. (1994). “Poultry Housing Tips: Environmental Controllers. University of Georgia Cooperative Extension Service.” College of Agricultural and Environmental Sciences, Athen, Georgia 30602, USA.

9. Eberhart R. C, Kennedy J. (1995). “A New Optimizer Using Particle Swarm Theory. In Proceedings of the 6th International Symposium on Micro-machine Human Science.” 1995: 39–43.
10. Engelbrecht A. P. (2005). “Fundamentals of computational swarm intelligence.” John Wiley & Sons, Ltd.
11. Kumar A., Singh B. K. & Patro B. D. K. (2016). “Particle Swarm Optimization: A Study of Variants and Their Applications.” International Journal of Computer Applications, 135(5): 24-30.
12. Ola B. O, Oguntoye J. P. and Awodoye O. O. (2017). “Performance Evaluation of Particle Swarm Optimization on Poultry House Temperature Control System.” IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN:2278-0661, p-ISSN:2278-8727. In press.
13. Rashedi E., Nezamabadi-pour H. & Saryazdi S. (2010). “BGSA: Binary Gravitational Search Algorithm”, Natural Computing, 9(3): 727-745.
14. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S. (2009). “GSA: A Gravitational Search Algorithm.” Information sciences, 179(13): 2232-2248.
15. Sinduja K., Sofia Jenifer S., Sri Abishek M. & Sivasankari B. (2016). “Automated Control System for Poultry Farm Based On Embedded System.” International Research Journal of Engineering and Technology (IRJET) 3(3): 620 -624.
16. Taisir E. & Al Qasim R. (2013). “On The Performance of the Gravitational Search Algorithm.” International Journal of Advanced Computer Science and Applications (IJACSA). 4(8): 74-78.
17. Tilakasiri, S. L. & Vidanapathirana, U. (1988). “Economics of poultry farming – a case study.” Economic Review, 13(9): 22-25.
18. William J., (1995), “Extension Poultry Specialist”, Iowa State University, Ames.
19. Zhang J., Y. Wang Y. & Feng J. (2013). “Parallel Multi-Swarm PSO Based on K-Medoids and Uniform Design.” Research Journal of Applied Sciences, Engineering and Technology, 5(8): 2576-2585.