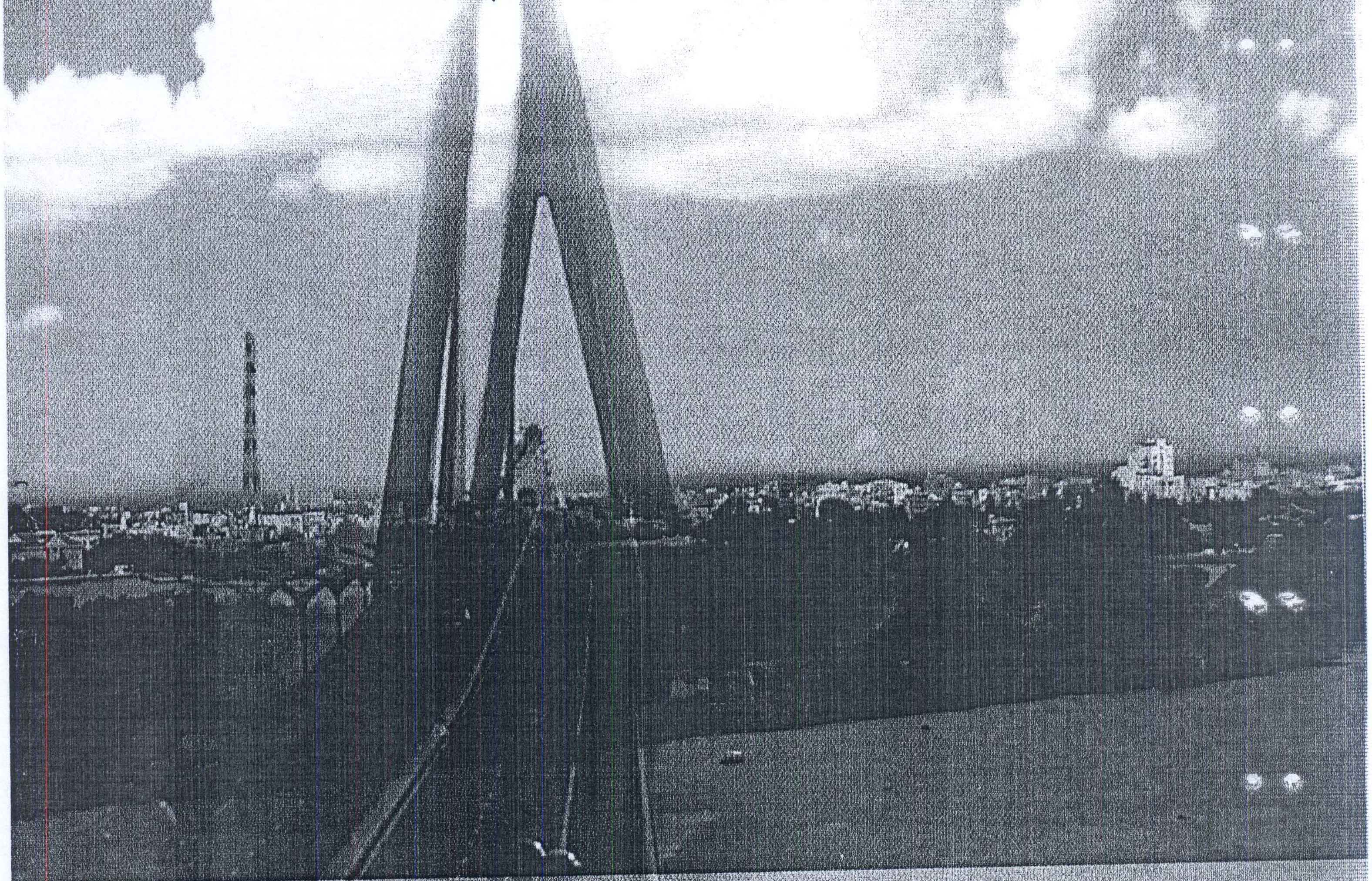
Bich-Thuy Dong Thi, Alexis Drogoul, Pierre Kuonen,
Cao-De Tran, Cong-An Tran, Nguyen Thai-Nghe (Eds.)

# The 2015 IEEE RIVF International Conference on Computing & Communication Technologies

## Research, Innovation, and Vision for Future (RIVF)

January 25-28, 2015, Can Tho University (CTU), Vietnam



# IEEE RIVF 2015 Proceedings

IEEE IEEE
Vietnam Section

# Table of Contents

## The 2015 IEEE RIVF International Conference on Computing & Communication Technologies

Research, Innovation, and Vision for Future (RIVF)

# Balance Particle Swarm Optimization and Gravitational Search Algorithm for Energy Efficient in Heterogeneous Wireless Sensor Networks

Trong-Thua Huynh[1,3], Anh-Vu Dinh-Duc[2], Cong-Hung Tran[3], Tuan-Anh Le[4]
[1]*HCM city University of Technology, Vietnam*
[2]*University of Information Technology, Vietnam*
[3]*Posts &Telecommunications Institute of Technology, Vietnam*
[4]*Faculty of Information Technology, Thu Dau Mot University, Binh Duong, Vietnam*
htthua@ptithcm.edu.vn, vudda@uit.edu.vn, conghung@ptithcm.edu.vn, letuanh@tdmu.edu.vn

*Abstract* – **In this paper, we propose a Balanced PSOGSA algorithm by combining the ability for social thinking in Particle Swarm Optimization with the local search capability of Gravitational Search Algorithm for reducing the probability of trapping in local optimum and prolonging time period before the death of the first node in wireless sensor network. Besides, we also improve the objective function to shorten the convergence time of the algorithm. The simulation results show that our proposed protocol has lower energy consumption and longer lifetime compared to other protocols.**

***Keywords* - *cluster, optimization, energy consumption.***

## I. INTRODUCTION

A new class of heuristic algorithms, inspired by behaviors of natural phenomena, is currently being developed that can potentially solve numerous problems of wireless sensor network (WSN). These algorithms rely on the communication of a massive amount of simultaneously interacting agents. These include Ant Colony Optimization (ACO) [1], Artificial Bee Colony (ABC) [2] and Particle Swarm Optimization (PSO) [3]. PSO was a popular multidimensional optimization technique that had been applied on many WSN problems over years [4], [5]. Although it was very simple but the global optimum solution was not comparable to the later methods. On the other hand, Gravitational Search Algorithm (GSA) [6] is a novel heuristic optimization method based on the law of gravity and mass interactions. It had also been applied on WSN [7], [8], giving better and more quality solutions but suffering from long execution time.

There are some studies have combined PSO with GSA to exploit both fast convergence and high quality optimum solutions from two mentioned methods such as PSOGSA for Function Optimization [9], PSOGSA for Nonconvex Economic Load Dispatch Problem [10], and PSOGSA for Training Feedforward Neural Networks [11]. However, to our knowledge, this approach has not been applied to WSN problems yet.

Clustering is the technique used very effectively to archive the energy efficiency in WSN. In clustering approach, sensor nodes elect themselves as cluster heads based on the probability values. However, it is obtained assuming that the sensor nodes are equipped with the same amount of energy, and as a result, they can not take full advantage of the presence of node heterogeneity. We are interested in the impact of heterogeneity in terms of node energy. In heterogeneous WSN, a percentage of the nodes is equipped with more energy than the rest of the nodes in the same network. As the lifetime of WSN is limited, it is necessary to re-energize the network by additional nodes. These nodes will be equipped with more energy than the nodes that are already in use, which creates heterogeneity in terms of node energy.

Motivated by above mentioned points, in this paper, we propose a Balanced PSOGSA algorithm (BPSOGSA for short) in heterogeneous WSN for reducing the probability of trapping in local optimum, prolonging the time interval before the death of the first node, and improving the objective function to shorten the convergence time.

The remainder of the paper is organized as follows. In section II, we present paticle swarm optimization algorithm and gravitational search algorithm. Section III describes our proposal. Section IV provides simulation results in terms of energy efficiency and number of dead nodes. Finally, concluding remarks are given in section V.

## II. PSO ALGORITHM AND GSA ALGORITHM

In this section, we provide a brief description of PSO algorithm and GSA algorithm.

### *A. PSO Algorithm*

Particle swarm optimization was developed by Kennedy and Eberhart in 1995 [3], based on the swarm behaviour of bird flocking in nature. Though particle swarm optimization has many similarities with genetic algorithms and virtual ant algorithms, but it is much simpler because it does not use mutation/crossover operators or pheromone. Instead, it uses the real-number randomness and the global communication among the swarm particles.

978-1-4799-8044-4/15/$31.00 ©2015 IEEE

In PSO, a set of potential solutions are called particles that are initialised randomly. Each particle will have a fitness value, which will be evaluated by the fitness function to be optimised in each generation. Each particle knows its best position *pbest* and the best position so far among the entire group of particles *gbest*. The particle will have velocities, which direct the flying of the particle. In each generation the velocity and the position of the particle will be updated [5]. In each iteration $k$, velocity $V$ and position $X$ are updated using (1) and (2). The update process is iteratively repeated until either an acceptable gbest is achieved or the maximum number of iterations is reached.

$$V_i^{k+1} = \omega V_i^k + c_1 r_1(pbest_i - X_i^k) + c_2 r_2(gbest_i - X_i^k) \quad (1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2)$$

where $\omega$ is inertia weight ($\omega = 0.5\frac{iter_{max}-iter}{iter_{max}} + 0.4$, where $iter_{max}$ is the maximum number of iterations allowed, *iter* is the current iteration. $\omega$ affects the convergence of PSO. Large $\omega$ facilitates global search, whereas small $\omega$ facilitates local search), $c_1$ and $c_2$ are constants, $r_1$ and $r_2$ are random numbers uniformly distributed in [0, 1], $V_i^k$ is velocity of particle $i$ at iteration $k$, $V_i^{k+1}$ is velocity of particle $i$ at iteration $k + 1$, $X_i^k$ is current position of particle $i$ at iteration $k$, $pbest_i$ is pbest of particle $i$, $gbest_i$ is gbest of the group, $X_i^{k+1}$ is position of the particle $i$ at iteration $k + 1$.

### B. GSA Algorithm

Gravitational Search Algorithm is a novel heuristic search algorithm developed by Rashedi et al. in 2009 [6]. The GSA is based on the Newton law of gravity and the law of motion. Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. In GSA, agents are considered as objects and their performance which will be calculated by using a fitness function expressed by their masses. In a system with $N$ agents (masses), the positions are defined as (3):

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

where $x_i^d$ presents the position of agent $i$ in dimension $d$.

At a specific time/iteration $(t)$, the force acting on mass $i$ from mass $j$ is defined as (4):

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{R_{ij}(k)+\varepsilon}\left(X_j^d(t) - X_i^d(t)\right) \quad (4)$$

where $M_{pi}$ is the passive gravitational mass related to agent $i$, $M_{aj}$ is the active gravitational mass related to agent $j$, $G(t)$ is gravitational constant at time $t$, $\varepsilon$ is a small constant, and $R_{ij}(t)$ is the Euclidian distance between two agents $i$ and $j$: $R_{ij}(t) = \|X_i(t), X_j(t)\|_2$. The $G(t)$ is calculated as (5):

$$G(t) = G_0 \times exp(-\alpha \times iter/maxiter) \quad (5)$$

Where $\alpha$ and $G_0$ are descending coefficient and initial value respectively, *iter* is the current iteration, and *maxiter* is maximum number of iterations.

In a problem space with the dimension $d$, total force that acts on agent $i$ is calculated as (6):

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j \times F_{ij}^d(t) \quad (6)$$

where $rand_j$ is a random number in the interval [0,1].

By the law of motion, the acceleration of the agent $i$ at time $t$ in direction $d$, $a^d_i(t)$ is given as (7):

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (7)$$

where $M_{ii}$ is the inertial mass of agent $i$.

The next velocity of an agent is considered as a fraction of its current velocity added to its acceleration:

$$V^d_i(t+1) = rand_i \times V^d_i(t) + a^d_i(t) \quad (8)$$

where $rand_i$ is a random variable in the interval [0,1].

When acceleration and velocity of each mass are calculated, the new position of the masses could be considered as (9):

$$X^d_i(t+1) = X^d_i(t) + V^d_i(t+1) \quad (9)$$

In GSA, at first all masses are initialized with random values. Each mass is a candidate solution. After initialization, velocities for all masses are defined using (8). Meanwhile the gravitational constant, total forces, and accelerations are calculated as (5), (6), and (7) respectively. The positions of masses are calculated using (9). Finally, GSA will be stopped by meeting the end criterion.

## III. BALANCED PSOGSA ALGORITHM

The our proposed protocol has three key characteristics. The first is combining the ability for social thinking in PSO with the local search capability of GSA to reduce the probability of trapping in local optimum. Secondly, It is a heterogeneous-aware protocol, in the sense that election probabilities are weighted by the initial energy of a node related to that of other nodes in the network. This prolongs the time interval before the death of the first node (we refer to as BALANCED). The third is improving the objective function by considering the position of the neighboring particles when updating the position of any particle in the network to shorten the convergence time of the algorithm. The algorithm is described as following steps:

Step 1: Network Initialization: initialize the parameters and network topology.

Step 2: In our protocol, sensor nodes are divided into advanced nodes and normal nodes. Advanced nodes have more energy and have to become cluster heads more often than the normal nodes. Choose advanced nodes and normal nodes based on fitness function as follows [12]:

$$C_{ij}(x,y) = \begin{cases} 0 & if\ r + r_e \leq d_{ij}(x,y) \\ e^{-\lambda\alpha^\beta} & if\ r - r_e < d_{ij}(x,y) < r + r_e \\ 1 & if\ r + r_e \geq d_{ij}(x,y) \end{cases} \quad (10)$$

where $d_{ij}(x,y)$ is Euclidean distance between sensor $s(x,y)$ and any grid point $(i,j)$; $r$ is detection range of each sensor node; $r_e$ is a measure of the uncertainty in sensor detection; $\alpha$, $\lambda$ and $\beta$ are parameters measuring detection probability when a target is at distance greater than $r_e$ but within a distance from the sensor.

Because advanced nodes consume more energy than normal nodes, assuming that $E_0$ is the initial energy of the normal node, the initial energy of advanced node will be $E_0(1 + a)$.

Step 3: Probability computation and cluster heads election. Probability of advanced node and normal node being cluster head are given by (11) and (12):

$$P_{nrm} = \frac{P_{opt}}{1+a\times m} \qquad (11)$$

$$P_{adv} = \frac{P_{opt}}{1+a\times m} \times (1 + a) \qquad (12)$$

where $P_{opt}$ is the desired percentage of cluster heads, $m$ is the fraction of advanced nodes and $a$ is the additional energy between advanced node and normal node.

A normal node becomes the cluster head in the current round with following threshold:

$$T(S_{nrm}) = \begin{cases} \frac{P_{nrm}}{1-P_{nrm}(r \bmod \frac{1}{P_{nrm}})} & if\ S_{nrm} \in G' \\ 0 & otherwise \end{cases} \qquad (13)$$

where $r$ is current round; $G'$ is the set of nodes that have not become cluster heads within the last $\frac{1}{P_{nrm}}$ rounds.

$T(S_{nrm})$ is threshold applied for $n\times(1-m)$ normal nodes. Thus, each round, there are $n\times(1-m)\times P_{nrm}$ normal nodes being cluster heads.

An advanced node becomes a cluster head in the current round with following threshold:

$$T(S_{adv}) = \begin{cases} \frac{P_{adv}}{1-P_{adv}(r \bmod \frac{1}{P_{adv}})} & if\ S_{adv} \in G'' \\ 0 & otherwise \end{cases} \qquad (14)$$

where $G''$ is the set of nodes that have not become cluster heads within the last $\frac{1}{P_{adv}}$ rounds.

$T(S_{adv})$ is threshold applied for $n\times m$ advanced nodes. Thus, each round, there are $n\times m\times P_{adv}$ advanced nodes being cluster heads.

So, there are $n\times(1-m)\times P_{nrm}$ + $n\times m\times P_{adv}$ nodes being cluster heads each round.

Step 4: Clustering based on distance from cluster head to remaining nodes:

- Calculate distance $d(n_i,CH_k)$ between node $n_i$ and all cluster heads.
- Assign node $n_i$ to cluster head $CH_k$ where:

$d(n_i, CH_k) = min\{d(n_i, CH_k)\}, i = 1,2...N, k = 1,2...C$

where $N$ is number of non cluster head; $C$ is number of clusters; $CH_k$ is cluster head of cluster $k$.

Step 5: Member nodes send data to cluster head and cluster head forwards the aggregated data to the base station. Consumed energy is given by (15):

$$E(L,d) = \begin{cases} L\times E_{elec} + L\times \varepsilon_{fs} \times d^2 & if\ d \le d_0 \\ L\times E_{elec} + L\times \varepsilon_{mp} \times d^4 & if\ d > d_0 \end{cases} \quad ....(15)$$

where $E_{elec}$ is the energy dissipated per bit to run the transmitter or the receiver circuit, $\varepsilon_{fs}$ and $\varepsilon_{mp}$ depend on the transmitter amplifier model that we use, $L$ is the message size, $d$ is the distance between the sender and receiver nodes and $d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$.

Step 6: Update accelerator for agent $i$ in dimension $d$ given by (16):

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \qquad (16)$$

where $Fi^d(t)$ is total force that acts on the agent $i$ in dimension $d$; $M_{ii}(t)$ is mass of inertia of agent $i$.

Step 7: In the PSO algorithm,, at a specific time, the velocity of an agent is updated by its own position and positions of all particles in the whole network but not rely on position of the neighboring agents. It significantly reduces the rate of convergence of the algorithm. To solve this, we modify the equation (1) as (17):

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (pbest_i - X_i^k) + c_2 r_2 (gbest_i - X_i^k) + c_3 r_3 (nbest_i - X_i^k) \qquad (17)$$

where $nbest$ is the best postion of node/agent compared to its neighboring nodes.

Hence, the next velocity of an agent is considered as a fraction of its current velocity added to the acceleration, combine (8) and (17), we have:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (a_i^t) + c_2 r_2 (gbest - X_i^t) + c_3 r_3 (nbest_i - X_i^k) \quad (18)$$

where $V_i^t$ is velocity of agent $i$ at the $t^{th}$ iterator; $r_1$, $r_2$, $r_3$ are random numbers uniformly distributed in [0, 1]; $c_1$, $c_2$, $c_3$ are constants; $a_i^t$ is acceleration coefficient of agent $i$ at the $t^{th}$ iterator; $gbest$ is the best solution for current.

Update velocity for agent $i$ by (18):

Step 8: Update position of agent $i$ by (19). Position and velocity will affect to clustering and energy consumption in next rounds.

$$X_i^{t+1} = X_i^t + V_i^{t+1} \qquad (19)$$

Repeat steps 2 through 8 until the maximum number of iterations is reached.

## IV. SIMULATION RESULTS

We simulate a clustered wireless sensor network for 100 nodes in a field with dimensions 1000m×1000m with unequal initial energy of nodes to show the effect of the different nodes' energy in the heterogeneous network. We set

10 percent of the total nodes to have 1 Joules of initial energy (i.e $a$ = 1 and $P_{opt}$ = 0.1) while 90 percent of remain nodes have 0.5 Joules of initial energy. Base station is located at (500, 500), the data message size was fixed at 500 bytes, maximum number of iterations = 100. We use parameters $c_1$ = 0.5 and $c_2$ = 1.5, $c_3$ = 1.0. We analyze performance evaluation compared with PSO and GSA in terms of the number of live nodes, total energy dissipated in the system over time.

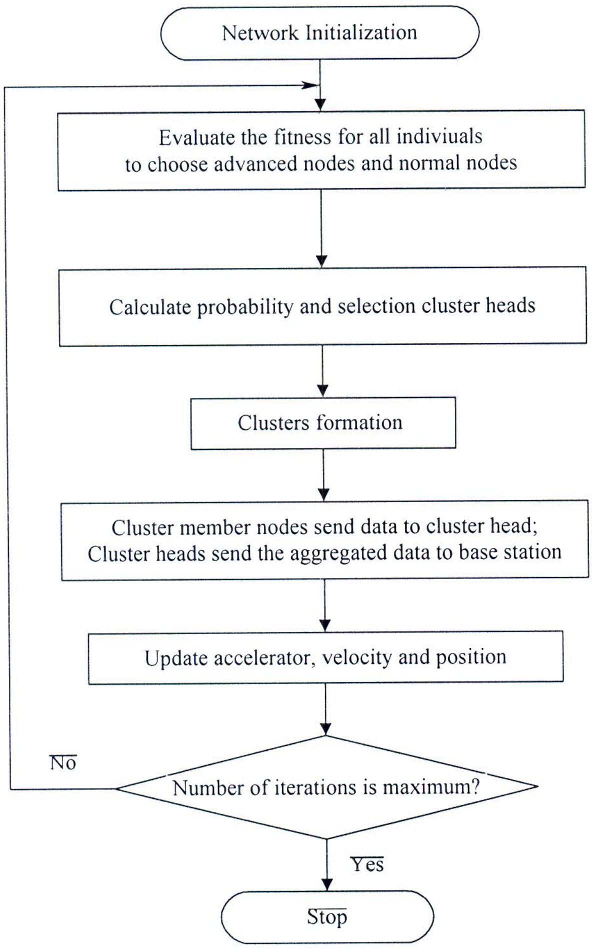



**Fig. 1** Flowchart of Balanced PSOGSA Algorithm.

Figure 2 shows the results of clustering for the 100 nodes network distributed randomly. Obviously, the proposed protocol *(c)* result in good network partitioning, where cluster heads are evenly positioned across the network and located near the centre of each cluster whereas PSO *(a)* and GSA *(b)* result in the uneven distribution of cluster heads throughout the sensor field.

Figure 3 illustrates the number of live nodes over time for the simulation with the base station located at the middle of the network area. Clearly, Balanced PSOGSA prolong the network lifetime significantly compared to PSO and GSA. This is because our improved algorithms minimize the distance between the non cluster head nodes to cluster head in each cluster while cluster heads are distributed optimally across the network.

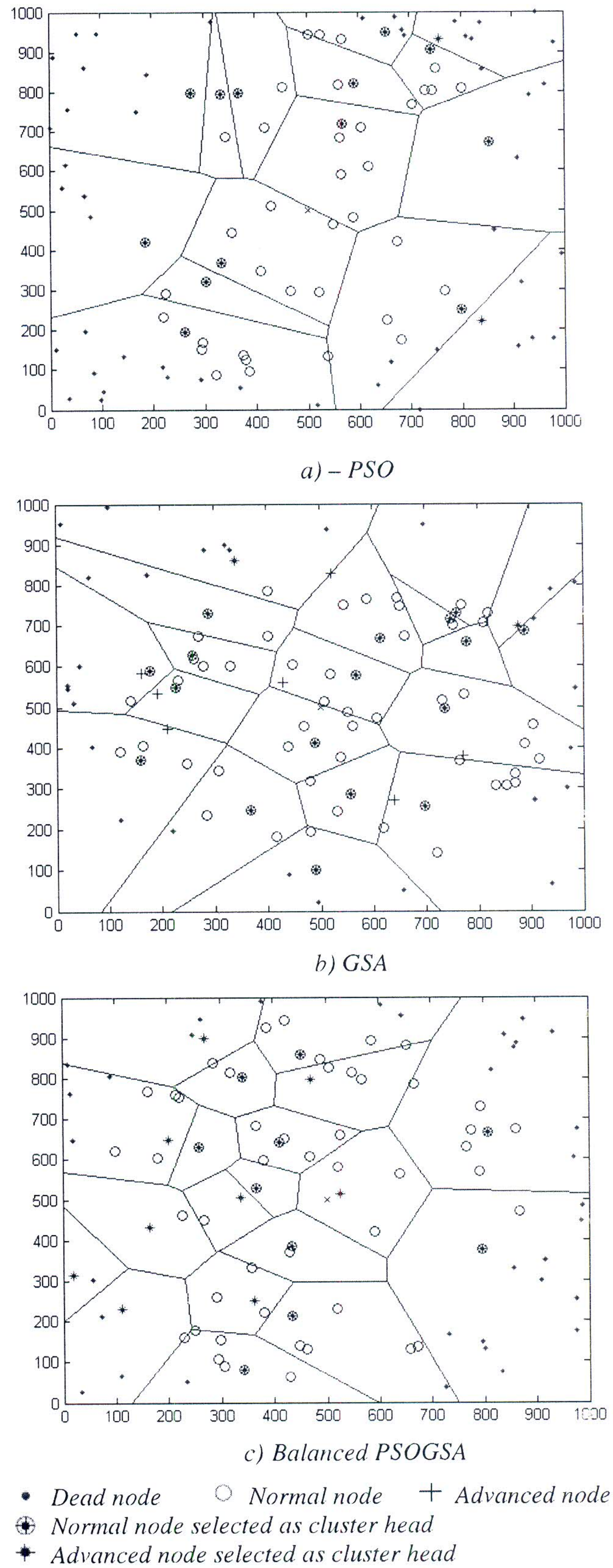



**Fig. 2** Results of network partitioning by different protocols.

Consequently, the total energy consumed by all nodes for communication is reduced since the distances between non-cluster head nodes and their cluster head are shorter. Figure 4 shows that the total consumed energy of the entire network by Balanced PSOGSA is less than by PSO and GSA over time.

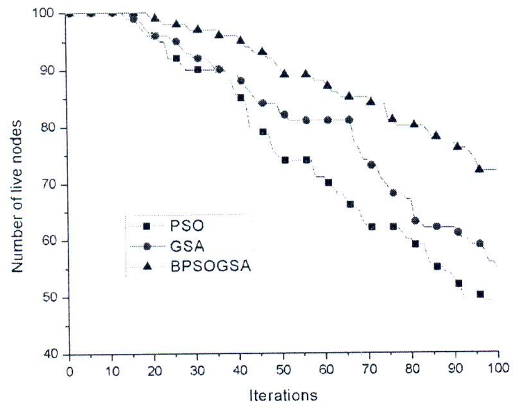



**Fig. 3** Number of live nodes over time.

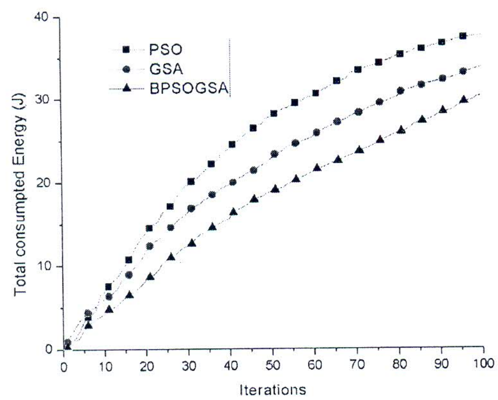



**Fig. 4** Total energy dissipated over time.

## V. CONCLUSIONS

In this reseach, we have proposed the Balanced PSOGSA algorithm in heterogeneous WSN for reducing the probability of trapping in local optimum, prolonging the time interval before the death of the first node, and improving the objective function to shorten the convergence time of the algorithm. Besides, our algorithm result in the better network partitioning where cluster heads are optimally distributed across the network by minimizing the distance between the cluster head and non cluster head nodes in each cluster. By simulation, we have shown the appropriateness of our algorithms evaluated againts PSO and GSA protocols in terms of the number of live nodes and the total consumed energy in the entire network over time.

In this paper, we used the objective function based on the distance between the nodes in each cluster for energy saving. However, in many applications that require the timely response of the system confronted with changes of the environment such as fire alarm systems, network delay is very important. Therefore, we are currently improving these algorithms to balance network delay and energy consumpion in WSNs.

## REFERENCES


[1] Ying Lin et al., "An Ant Colony Optimization Approach for Maximizing the Lifetime of Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Human-Machine Systems*, vol. 42, no. 3, pp. 408-420, 2011.

[2] Dervis Karaboga et al., "Cluster based wireless sensor network routing using artificial bee colony algorithm," *ACM Journal Wireless Networks*, vol. 18, no. 7, pp. 847-860, 2012.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE International Conference on Neural Networks*, 1995.

[4] Singh B and Lobiyal D.K, "A novel energy-aware cluster head selection based onparticle swarm optimization for wireless sensor networks," *SpingerOpen Journal*, 2012.

[5] Raghavendra V. Kulkarni and Ganesh Kumar Venayagamoorthy, "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, no. 2, 2011.

[6] E.Rashedi, H.Nezamabadi-pour and S.Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences Elsevier*, p. 2232–2248, 2009.

[7] M.K Rafsanjani and M.B Dowlatshahi, "A Gravitational Search Algorithm for Finding Near-optimal Base Station Location in Two-Tiered WSNs," in *International Conference on Machine Learning and Computing*, 2011.

[8] J. Rejina Parvin and C. Vasanthanayaki, "Gravitational Search Algorithm based Mobile Aggregator Sink nodes for energy efficient Wireless Sensor Networks," in *International Conference on Circuits, Power and Computing Technologies*, 2013.

[9] Seyedali Mirjalili et al., "A New Hybrid PSOGSA Algorithm for Function Optimization," in *International Conference on Computer and Information Application*, 2010.

[10] Mani Ashouri et al., "Application of New Hybrid Particle Swarm Optimization and Gravitational Search Algorithm for Non Convex Economic Load Dispatch Problem," *Journal of Advances in Computer Research*, vol. 4, no. 2, pp. 41-51, 2013.

[11] SeyedAli Mirjalili et al., "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Applied Mathematics and Computation Journal of Elsevier*, vol. 218, no. 22, p. 11125–11137, 2012.

[12] W. Xiaoling et al., "Energy-efficient Deployment of Mobile Sensor Networks by PSO," *Advanced Web and Network Technologies, and Applications*, pp. 373-382, 2006.