Applied Soft Computing 55 (2017) 452–461

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

HACH: Heuristic Algorithm for Clustering Hierarchy protocol in wireless sensor networks☆

Muyiwa Olakanmi Oladimeji*, Mikdam Turkey, Sandra Dudley

School of Engineering, London South Bank University, 103 Borough Road, London SE1 0AA, United Kingdom

ARTICLE INFO

Article history:
Received 23 June 2016
Received in revised form 13 January 2017
Accepted 15 February 2017

Keywords:
Wireless sensor networks
Sleep scheduling
Clustering
Heuristic crossover
Coverage
Energy heterogeneity

ABSTRACT

Wireless sensor networks (WSNs) require energy management protocols to efficiently use the energy supply constraints of battery-powered sensors to prolong its network lifetime. This paper proposes a novel Heuristic Algorithm for Clustering Hierarchy (HACH), which sequentially performs selection of inactive nodes and cluster head nodes at every round. Inactive node selection employs a stochastic sleep scheduling mechanism to determine the selection of nodes that can be put into sleep mode without adversely affecting network coverage. Also, the clustering algorithm uses a novel heuristic crossover operator to combine two different solutions to achieve an improved solution that enhances the distribution of cluster head nodes and coordinates energy consumption in WSNs. The proposed algorithm is evaluated via simulation experiments and compared with some existing algorithms. Our protocol shows improved performance in terms of extended lifetime and maintains favourable performances even under different energy heterogeneity settings.

1. Introduction

Recent progress in wireless communications and microelectronics have contributed to the development of sensor nodes that are agile, autonomous, self-aware and self-configurable. These sensor nodes are densely deployed throughout a spatial region in order to sense particular event or abnormal environmental conditions such as moisture, motion, heat, smoke, pressure, etc. in the form of data [1]. These sensors, when in large numbers, can be networked and deployed in remote and hostile environments enabling sustained wireless sensor network (WSN) connectivity. Hitherto WSNs have been used in many military and civil applications, for example, in target field imaging, event detection, weather monitoring, tactile and security observation scenarios [2]. Nevertheless, sensor node distribution and network longevity are constrained by energy supply and bandwidth requirements. These noted constraints mixed with the common deployment of large numbers of sensor nodes must be considered when a WSN network topology is to be deployed. The design of energy efficient scheme is a major challenge especially in the domain of routing, which is one of the key functions of the WSNs [3]. Therefore, inventive techniques which reduce or eliminate energy inadequacies that would normally shorten the lifetime of the network are necessary. In this paper, the authors present a method which balances energy consumption among sensor nodes to prolong WSN lifetime. Energy resourcefulness is uniquely obtained using two described mechanisms; firstly, cluster head (CH) selection using a generic algorithm (GA) is employed that ensures appropriately distributed nodes with higher energies will be selected as CHs. Secondly, a Boltzmann inspired selection mechanism was utilized to select nodes to send into sleep mode without causing an adverse effect on the coverage.

The commonest routing protocols deployed to address the challenges discussed above are generally categorised into two classes, namely flat and hierarchical. Flat protocols comprise the well-known Direct Transmission (DT) and Minimum Transmission Energy (MTE), which do not provide balanced sensor energy distributions in a WSN. The disadvantage of the MTE is that a remote sensor normally employs a relay sensor when transmitting data to/from the sink and this results in the relay sensor being the first node to die. In the DT protocol, the sink communicates directly with sensors and this results in the death of the remote sensor first. Consequently when creating WSNs, energy-efficient clustering protocols act as a pivotal factor for sensor lifetime extension. Generally, clustering protocols can perform better than flat protocols in terms of balancing energy consumption and network

---

☆ This paper is an extended, improved version of the paper (A heuristic crossover enhanced evolutionary algorithm for clustering wireless sensor network) presented at EvoComNet2016 and published in: Applications of Evolutionary Computation, 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings, Part I, LNCS 9597, pp. 251–266, Springer, 2016.

* Corresponding author.

E-mail addresses: oladimm2@lsbu.ac.uk (M.O. Oladimeji), turkeym@lsbu.ac.uk (M. Turkey), dudleyms@lsbu.ac.uk (S. Dudley).

http://dx.doi.org/10.1016/j.asoc.2017.02.016

lifetime prolongation by employing data aggregation mechanisms [4,5]. In WSNs, there are three types of nodes considered: the cluster-head (CH), member node (MN) and sink node (SN). The member node manages sensing of the raw data and utilizes Time Domain Multiple Access (TDMA) scheduling to send the raw data to the CH. The CH must aggregate data received from MNs and forward the aggregated data to the SN through single-hop or multi-hop. CH selection can be carried out by the sensors individually, by the SN or can be pre-implemented by the wireless network designer. Here, CH selection is performed by the SN due to the fact that the SN has sufficient energy and can perform multifaceted calculations. The problem of CH selection can be considered as an optimization issue where the methods have employed GA to solve. Here the authors define an objective function that evaluates the discrete solution and propose an innovative heuristic crossover which is enhanced by the knowledge of our problem.

In this paper, we present a new Heuristic Algorithm for Clustering Hierarchy (`HACH`) protocol that simultaneously performs sleeping scheduling and clustering of sensor nodes upon each round. For sleep scheduling operation, the authors have developed the stochastic selection of inactive nodes (`SSIN`). A protocol that imitates the Boltzmann selection process in GA was used to decrease the number of active nodes in each round by putting some nodes to sleep or into inactive mode so that energy could be conserved and network lifetime increased without harming coverage. We further developed the Heuristic-Crossover Enhanced Evolutionary Algorithm for Cluster Head Selection (`HEECHS`) protocol for the clustering operation. `HEECHS` uses the known information around the problem to develop a useful heuristic crossover that combines genetic material in a unique way to produce improved CH configuration. This method described has some parallels with optimization algorithms known as Memetic Algorithm (MAs). This algorithm is a type of stochastic global search heuristics in which Evolutionary Algorithm-based techniques are mixed with a local search technique to improve the quality of the solutions proposed by evolution [6]. Sleep scheduling and clustering algorithms work together to optimize network lifetime by harmonizing energy consumption amongst sensor nodes during the communication times. Energy consumption optimization is performed by selecting spatially distributed nodes with higher energy as CHs and additionally placing certain nodes into sleep mode without harming coverage. The `HACH` protocol proposed performs very well compared to protocols that use GA because it integrates knowledge of the problem into GA crossover operator.

The rest of the paper is organised as follows. Section 2 presents related work on energy conservation techniques and clustering protocols in the area of energy-efficient wireless sensor networks. Section 3 describes the network and radio model assumptions that underlie the protocol presented. In Section 4 the authors describe our proposed algorithm under three pivotal operational phases, those being the sleep scheduling mechanism, clustering algorithm and the energy consumption calculation. Section 5 presents our experimental set-up, performance procedures, results and discussion. Finally, Section 6 provided our conclusion.

2. Related work

In WSN environments, sensor node sleep scheduling can be used as an energy conservation method for network lifetime extension. In [7], a coverage maximization with sleep scheduling protocol (CMSS) that ensures network areas are fully covered by selected active sensors was presented. Each sensor exchanges information with its neighbouring sensors and sets a waiting time. During sensor waiting times, a sensor can receive a sleep message from neighbouring nodes. When a sensor receives these messages, it updates its own neighbour and cell value table. If the minimum value of the cell value table of a sensor equals to one, it silently becomes an active node. Otherwise, it will wait for the waiting time to expire before it turns into an inactive node. An energy preserving sleep scheduling (EPSS) strategy allows each sensor to make decisions regarding going into sleep mode based on their distance from the cluster head and network density. This guarantees balanced energy consumption in the cluster by taking into account the density of node deployment and the network load while determining the sleep probability [8]. In [9], a probabilistic and analytical method was employed to approximate the overlapping sensing coverage between a node and its neighbours. It also estimates when a node can be put into sleep without jeopardizing expected coverage. The method is employed by the proposed scheduling and routing scheme to diminish control message overhead while considering the next mode (full-active, semi-active, inactive/sleeping) of sensor nodes.

Apart from energy conservation techniques, energy-efficient clustering protocols can also be employed to reduce and balance energy consumption across sensor nodes in WSNs to prolong lifetime [10–13]. At the time of CH and non-CH selection, the Low-Energy Adaptive Clustering Hierarchy (LEACH) assumes that the energy of each sensor node is the same. The selection process is carried out probabilistically and the CH's main role is to aggregate the data received from its cluster members and transmit the aggregated data directly to the sink. Difficulties with this protocol arise because the location of the selected CH may be some distance from the sink, thus it will consume more energy when transmitting to the sink. This can then result in CH nodes dying faster than other nodes [5]. A two-level LEACH (TL-LEACH) described in [14], adds an extra level to the cluster whereas LEACH has only one level. This additional level diminishes energy consumption particularly for CHs quite a distance away from the sink. The hybrid energy efficient distributive (HEED) protocol proposed in [15] selects CHs by employing residual energy and the least amount of energy used for communication between the CHs and non-CHs. The sink accepts data from the nodes using a multi-hop communication approach.

In the proposed Topology-Controlled Adaptive Clustering (TCAC) protocol [16], many nodes can consider themselves CH node candidates and inform other nodes of this. Every candidate CH node then examines if the other candidate CH nodes have a higher residual energy level or not. If there are none with higher residual energy, the highest announces itself the CH. The CH which has the minimum-cost distance between itself and the CH to the sink is selected by non-CHs. The size of the cluster is balanced by the TCAC protocol and data is then sent directly to the sink from the CH. Within the proposed scalable energy efficient clustering hierarchy (SEECH) protocol [17], network nodes are separated into three layers, those being the member nodes, CH nodes and relays. Clusters evolution is based on how central the CH node is with minimum intra-cluster energy distribution. A node close to the sink in a cluster is often selected as the relay node. The CH node is assisted by the relay node to transmit aggregated data to the sink through hop or multi-hop communication. A genetic algorithm based energy efficient cluster (GABEEC) protocol was described in [18]. Here clustering with dynamic CH selection was employed. An associate member node becomes a CH at the end of each round with this decision based on the remaining energy of the current CHs and the average energy of cluster members. The genetic algorithm approach was described and was aimed to diminish communication distances and optimize network lifetime. Another paper discussed a centralized energy-aware cluster-based protocol to extend the network lifetime of sensors by employing Particle Swarm Optimization (PSO) algorithm in [19]. The authors also defined a new cost function that simultaneously accounts for the maximum distance between the non-CH node and its CH, and

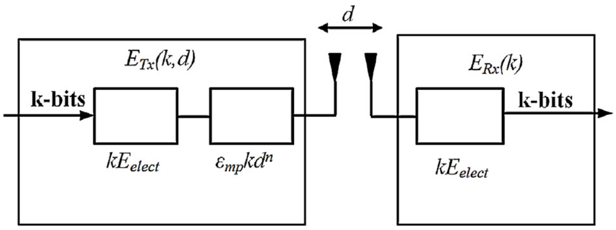


Fig. 1. Radio energy dissipation model.

the remaining energy of CH candidates in the CH selection algorithm.

3. Network and radio model assumptions

In the HACH protocol proposed, important network and radio model assumptions are presented as follows:

- The data sink is a stationary and resource-rich device that is placed far away from the sensing field.
- All sensors are stationary after deployment and average energy is constant in either homogeneous or heterogeneous environment.
- All sensors have GPS or other location determination devices attached to them. Hence, the HACH algorithm cannot be deployed for GPS-free sensor applications.
- Nodes are able to perform in inactive mode or a low power sleeping mode.
- Nodes that are close to each other have correlated data.
- The communication channel considered is assumed symmetric (i.e. the energy needed to transmit data from sensor node s1 to sensor node s2 is equal to the energy required to transmit a message from node s2 to node s1 for a particular signal to noise ratio (SNR)).

To ensure just comparison with previous protocols [5,20,21], the authors have employed the simple model for the radio hardware energy dissipation where the transmitter dissipates energy $E_{Tx}(k, d)$ to manage the radio electronics and the power amplifier, and the receiver dissipates energy $E_{Rx}(k)$ when managing the radio electronics, as shown in Fig. 1. The free space (d^2 power loss) and the multipath fading (d^4 power loss) channel models were used (depending on the distance (d) between the transmitter and receiver) for all the experiments described. The power-amplifier is fittingly managed so that should the distance be less than a threshold distance, we employ the free space (fs) model; else, the multipath (mp) model is used. Thus, to transmit a k-bit message a distance d, the radio spends:

$$E_{Tx}(k,d) = \begin{cases} kE_{elect} + \varepsilon_{mp}kd^4, & \text{if } d > d_0 \\ kE_{elect} + \varepsilon_{fs}kd^2, & \text{if } d < d_0 \end{cases} \tag{1}$$

And to receive k-bit message, the radio uses:

$$E_{Rx}(k) = kEelect \tag{2}$$

where the equation $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$ signifies the threshold distance and the electronics energy, factors such as the digital coding, modulation employed as well as filtering, and spreading of the signal effect E_{elect}. The amplifier energy, ε_{mp} or ε_{fs} depends on the distance to the receiver and the acceptable bit-error rate.

4. The proposed HACH protocol

There are three consecutive operations within the proposed protocol: sleep scheduling, clustering and network operations. The sink transmits control packets at the initial set-up phase so that it can receive node information in terms of the nodes ID, location and energy. The SSIN protocol proposed dynamically selects the nodes to send to sleep by generating an initial candidate list. This list is populated with nodes having lower energies than the average energy of all the nodes. Employing a stochastic process, a small number of nodes are subsequently placed into sleep mode without harming coverage. CH selection employing HEECHS is then completed on the remaining active nodes.

The proposed HEECHS protocol operates at the network layer of WSNs layered model presented in [22], which is similar to the Open System Interconnection (OSI) network model. After nodes deployment, the sink transmits and receives control packets containing the coordinates and energy value of all nodes. Using the obtained sensor coordinates, the sink computes the Euclidean distances between two adjacent nodes and each node to the sink. Each sensor is updated with the computed Euclidean distances between itself, its neighbours and the sink. These Euclidean distances and energy values are both used in establishing the cluster-based network topology for the purpose of packet routing.

Algorithm 1. Proposed HACH protocol

```
Let AliveNodes be the total number of sensor nodes
Compute the network total coverage.
while (AliveNodes > 0) do
  Use algorithm SSIN to select inactive nodes. (See Algorithm 2)
  Put selected nodes into sleep mode.
  Apply the proposed HEECHS algorithm for CHs configuration. (See Algorithm 3)
  Compute the energy values of $E_{CH}$, $E_{Mem}$ and $E_{Res}$. (refer to Section 4.3.3)
  Calculate the number of dead nodes (node with energy equal or less than 0).
  Update AliveNodes.
end while
```

Here, the authors have considered clustering as an optimization problem which would be best accomplished using GA. Tournament selection, mutation operator and the heuristic crossover are the genetic operators used in this approach. The most suitable CH configuration which guarantees balanced energy consumption across the network topology is selected at every network operation round. The residual energy of each node is calculated at the end of each round. This computed value is then employed to calculate the average energy for the next round. This cycle subsequently repeats until all network nodes are dead, as shown in Algorithm 1.

4.1. Sleep scheduling mechanism

In this section, we discuss the estimation of coverage by setting up a matrix that computes the number of nodes covering the area within each grid point. Furthermore, we present our SSIN protocol that uses the energy values and coverage effect in deciding which nodes to send into sleep mode.

4.1.1. Coverage estimation and matrix setup

Coverage is estimated by dividing the sensing field into uniform grid areas. The number of sensors that cover each point on the grid is computed by calculating the euclidean distance between each grid point and the individual sensor's point using their coordinates. If the euclidean distance between the two points is within the sensing range R_s, the point is taken to be covered by the sensor. The coverage matrix in Fig. 2 helps to identify the grid points that are not covered by any sensor and the points covered by one or more sensors.

4.1.2. Inactive node selection using SSIN mechanism

Conclusions as to which nodes to send into inactive mode at the beginning of each network operation round is made by the SSIN. The sleeping nodes candidate list evolves through the inspection of

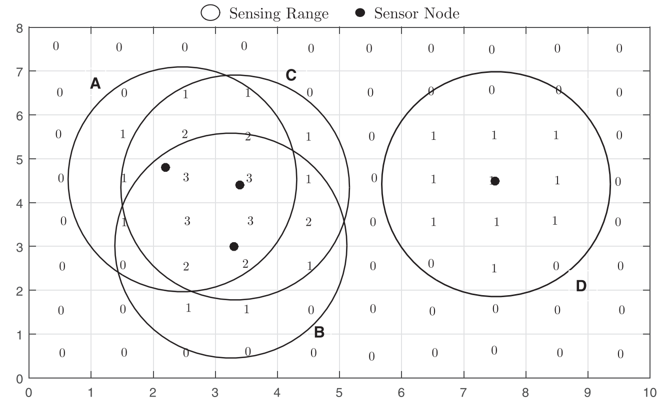


Fig. 2. Coverage matrix of covered grid points by sensors in 10 × 8 sensing field.

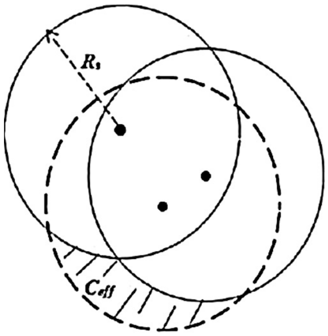


Fig. 3. Illustration of nodes to sleep on coverage area.

which nodes have residual energy less than the computed average energy. This selection process is tantamount to the Boltzmann selection process whereby a method is adopted to control the selection pressure [23]. The temperature parameter is varied in the Boltzmann selection process to effectively control the selection pressure. The maximum coverage effect, Max_{eff} is employed in this paper to regulate the effect of putting WSN nodes to sleep and is defined as:

$$Max_{eff} = 2\pi R_s^2 \tag{3}$$

Here, R_s is the range over which a sensor node senses (taking the coverage area as a circle with radius R_s), $(pi \times R_s^2)$ is the coverage of one node and the value '2' represents coverage of two nodes.

The coverage effect C_{eff} as shown in Fig. 3, is the effect of putting a node to sleep based on coverage. The total coverage effect is computed by summoning a matrix called the Coverage Matrix. This matrix captures node coverage areas that overlap permitting the identification of nodes that can be placed into sleep mode without harming coverage as there will be other nodes covering the selected node's area. The accumulated Coverage effect Acc_{eff} is defined as the total effect on the coverage as a result of allowing some nodes to sleep. Our algorithm presented here has been created to ensure the Acc_{eff} value is expected to be less than the Max_{eff} for optimum coverage ($Acc_{eff} < Max_{eff}$). The probability that a node will be added to the sleeping node list can be computed using:

$$P = e^{(-C_{eff}/Max_{eff})/(1-(Acc_{eff}/Max_{eff}))^2} \tag{4}$$

where the Acc_{eff} is the value to be minimized and Max_{eff} is a control parameter analogous to temperature in the Boltzmann tournament selection [24]. The computed probability, P is compared to a randomly generated number in the range [0,1], uniformly at random. An inactive node candidate list is formed stochastically if the *random*(0, 1) is less than P. Acc_{eff} is calculated by adding its current value to the C_{eff} value. The `SSIN` operation continues until Acc_{eff} is larger than Max_{eff} as described in Algorithm 2.

Algorithm 2. Proposed `SSIN` protocol

```
$Acc_{eff} = 0$;
Compute the residual energy, $E_{Res}$ of each node. (refer to Section 4.3.3)
Compute the average energy of all nodes, $E_{Avg}$.
Generate a candidate list for nodes that satisfies the condition $E_{Res} < E_{Avg}$.
Compute $Max_{eff}$. (refer to Eq. (3))
while ($Acc_{eff} < Max_{eff}$) do
  Compute probability, $P$ of adding nodes to the sleeping list. (See Eq. (4))
  if ($random(0, 1) < P$) then
    Create list of sleeping node from the candidate list.
    Compute the coverage effect, $C_{eff}$.
    $Acc_{eff} = Acc_{eff} + C_{eff}$
  end if
end while
```

4.2. *Clustering operations using* `HEECHS` *protocol*

The clustering operation is divided into stages: CH selection, cluster formation, data aggregation and data communication. As shown in Fig. 4, the setup state starts by the CH selection stage and proceeds by cluster formation. The setup state is followed by the data transmission state, which is subdivided into data aggregation and data transmission phases. During the setup state, a sink-assisted clustering algorithm that performs CH selection and membership association is applied to the active nodes in the network. During network initialization, sensors send their energy and location information to the sink in order to implement the proposed algorithm. The `HEECHS` protocol favours the selection of a CH that has higher energy and is far from neighbouring CH. As illustrated in Fig. 5, an energy efficient sensor node distribution is constructed by our proposed algorithm at every network operation round. Sensors are assigned to the closest CH thereby forming a single cluster. A TDMA schedule is assigned for each cluster to schedule packet transmission to that CH by the member nodes. All the information about clusters and TDMA schedule packets is broadcast to the entire

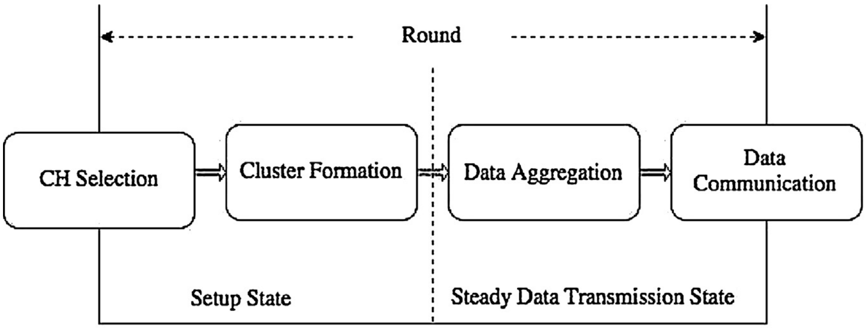


Fig. 4. One round of the clustering process.

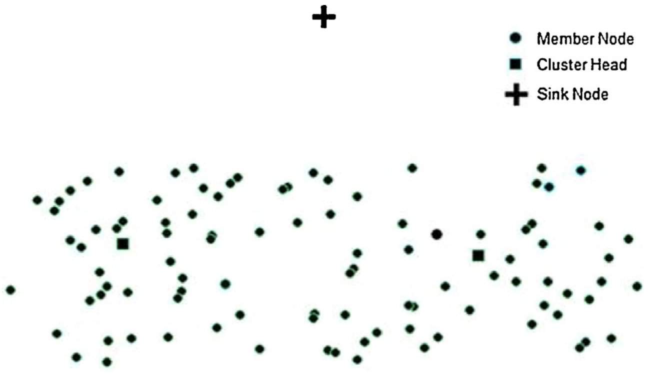


Fig. 5. Sensor nodes topology and random distribution.

network. Based on the time slot in the TDMA schedule packets, each node in a cluster send sensed data to their respective CH.

At each round, the sink performs a re-clustering procedure to form a new cluster-based topology that preserves the WSNs coverage and energy efficiency characteristics by rotating the CH role among sensors with scalability of hundreds to thousands. Scalability implies that there is a need for balanced energy consumption among the sensor nodes during communication through an efficient clustering algorithm [25]. The CH loses energy faster than the member nodes; hence the need for re-clustering or rotating the CH role among sensors in order to balance the energy consumption. Re-clustering is performed at the end of a round, which is the total time span for processes involved in the setup and steady data transmission state. The time-length of each round must be carefully decided because a large time length drains CHs energy and a short time-length result into overhead caused by frequent re-clustering [26]. The round time-length of our proposed algorithm adjust itself dynamically based on the number of active nodes in the WSNs.

In this work, the HEECHS protocol proposed is developed for the CH selection task using a heuristic-based GA. It runs through a number of tasks, similar to conventional GAs, such as population strings creation, string evaluation, best string selection and finally reproduction to create a new population. The unique, but significant difference is that the HEECHS protocol employs a problem-dependent knowledge-based heuristic crossover to find the best CH configuration with the optimum number of appropriately distributed CH nodes. In the proposed HEECHS, the genetic process of finding the best solution is performed using an energy unlimited sink device that can handle high execution time complexity and computation. The individuals within population $P(t)$ are coded by $0-1$ binary representation where '0' denotes a member node and '1' denotes a CH node as shown in Fig. 6.

Each individual with length N_s in a population size p_s is evaluated by computing the fitness value using Eq. (6). Individuals with the best fitness value are selected from two randomly selected parent pairs, $P(x)$ and $P(y)$. This process continues until the mating pool is filled. The heuristic crossover proposed here is subsequently applied to the individuals in the pool and a new population $P(t+1)$ is produced. Again, each individual fitness value in this new population is computed using Eq. (6) and the entire cycle continues until the stopping criterion is achieved. The stopping criterion is realized when the populations average fitness undergoes no further changes.

4.2.1. Proposed objective functions

To solve the CH selection problem, objective functions are developed because CH selection is considered an optimization problem. These objective functions return fitness values which are employed to assess the quality of a candidate solution. An objective function is found by taking into account parameters such as the total sensor node energy and the Risk penalty R. The sensor node energy parameter is considered to ensure that nodes with greater energy are given higher priority in the CH selection process.

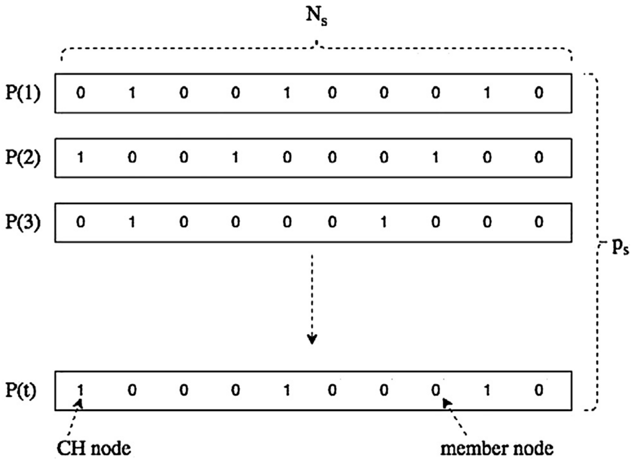


Fig. 6. Binary representation of individuals in the population.

The Risk penalty, R for the CH selection is defined as:

$$R=\begin{cases} Lower-L, & \text{if } \ L<Lower \\ L-Upper, & \text{if } \ L>Upper \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Based on many iterative tests, the percentage of CHs number (L) to the total number of sensor nodes (n) in the field always results in an optimal result between a *Lower* limit of 4% and *Upper* limit of 6%. Restrictions are imposed on the number of CHs using the parameter R.

Subsequently, the objective function is computed using:

$$F(X)=w_1*\frac{AvgENCH}{AvgECH}+w_2*R \tag{6}$$

where w_1 and w_2 are the weighting factors. The average energy of non-CHs, *AvgENCH* is the energy summation of all member nodes divided by the total number of member nodes $(n-L)$ as given below:

$$AvgENCH=\frac{\sum_{i\varepsilon NCH}E_i}{n-L} \tag{7}$$

Also, the average energy of CHs, *AvgECH* is the energy summation of all CH nodes divided by the total number of CHs (L) as given below:

$$AvgECH=\frac{\sum_{i\varepsilon CH}E_i}{L} \tag{8}$$

In Eq. (6), the ratio *AvgENCH*/*AvgECH* is given a higher weighting factor ($w_1=0.9$) than the Risk penalty, R ($w_2=0.1$) because of its importance. (Note: *CH* and *NCH* represent the set of all CHs and non-CHs respectively).

4.2.2. Proposed heuristic crossover

The principal operator used in the HEECHS protocol to produce new solutions is the heuristic crossover. This is a problem-dependent crossover that utilizes knowledge of a problem to fuse two potential resolutions, producing a new solution. According to Lixin Tang [27], a heuristic crossover is an operator that makes use of parents' inherent information to produce an offspring. In the canonical approach, individuals in a population are selected and two parent individuals are combined using the crossover operator to produce a pair of offspring that will replace its parents. Correspondingly, there is no assurance that an offspring would be superior to its parents in the canonical approach [28]. Contrarily, the heuristic crossover operator generates only one offspring from two or more parents and it is certain that the offspring would be of higher quality than the parents. As shown in Algorithm 3, the proposed heuristic crossover generates a single solution with CHs that are spatially distributed in the sensor field and selects nodes with higher energy to be the CH.

Algorithm 3. Proposed heuristic crossover

```
Select two individuals from the parent population.
Compute and keep the CH position in each individual in $CH_1$ and $CH_2$.
Compute the threshold distance, T (refer to Section 4.2.2)
Compute the union set $CH_{all} = CH_1 \cup CH_2$
Obtain the first CH position $CH_{all}(1)$ in the $CH_{all}$ set.
Generate a new set $CH_{new}$ and transfer the $CH_{all}(1)$ to it.
Compute the distance, $D$ between CH positions in the sets $CH_1$ and $CH_2$.
while ($D < T$) do
  if ($CH_{all}$ node energy < $CH_{new}$ node energy) then
    Discard the CH node. (i.e. do not add to $CH_{new}$ set)
  end if
  Replace the CH in the $CH_{new}$ set
end while
Add to the CH in the set $CH_{all}$ into the $CH_{new}$ set.
```

The CH genes position in each individual of selected parent pair is computed. An array that holds the genes position in both parent pairs is expressed by CH_1 and CH_2. We decided to define the threshold distance between any two adjacent CH position as $\frac{\sqrt{(x_{\max}-x_{\min})^2+(y_{\max}-y_{\min})^2}}{n\times 0.04}$, where the ($x_{min}$, y_{min}) and (x_{max}, y_{max}) coordinates represent the minimum and maximum xy points in the sensing field, ($n \times 0.04$) indicates 4% of all sensor nodes. A set CH_{all} is generated from the union of CH1 and CH2 (refer to Algorithm 3). The first CH position in the union set CH_{all} is moved into a new set CH_{new} by default. As shown in Algorithm 3, the decision to move successive CH positions from the CH_{all} to CH_{new} is based on spatial distance between CHs and residual energy.

4.2.3. *Other operators*

The efficacy of a genetic algorithm relies upon maintaining a balance between the concept of exploration and exploitation. Exploration is provided by crossover and mutation while selection enables exploitation [29,30]. The rest of the operators used in our proposed `HEECHS` protocol are discussed below:

- The *Tournament selection operator* selects individuals with the best fitness from groups of individuals randomly chosen from the current population. The selection pressure depends on the tournament size of the operator. In order to reduce the selection pressure, a tournament size of two was used for our algorithm and this process continues until the mating pool is full.
- The *Mutation operator* changes an individual (parent) with a mutation probability (pm) to produces one individual (offspring) with new fitness value.

The parent and child individuals in the initial population pool produced in the previous step are arranged in ascending order based on their fitness value. Subsequently, individuals with minimum fitness values are selected and they form the next generation's population. The *stopping criterion* is achieved when there is no further change in the fitness value of the population.

4.3. *Network operations and energy consumption computation*

In this algorithm, the network operations is divided into the set-up and steady phase. At each round the energy consumption value is computed by examining what happens to each node during both phases.

4.3.1. *Set-up phase*

The sink transmits and receives control packets from all nodes during the set-up phase in order to initiate the inter- and intra-communication. This control packets k_{CP} contain short messages that wake up and requests IDs, positions and energy level from all sensor nodes. As in Eq. (2), the energy $E_{Rx}(k_{CP})$ is spent to receive control packets from the sink. Also in Eq. (1), all nodes use energy $E_{Tx}(k_{CP}, d)$ transmitting control packets containing information about their IDs, positions and energy levels to the sink. The sink processes control packets and certain decisions are made, such as which nodes to send into sleep mode, which nodes will become CH and the membership association of each CH. All nodes also use energy $E_{Rx}(k_{CP})$ to receive their status information (whether CH or members) from the sink. The energy spent by all CHs to send TDMA schedules to their members is given as:

$$E_{Tx(ch_i)}(k_{CP}, d_{i-toMem}) = \sum_{i=1} ch_i * \begin{cases} k_{CP}E_{elect} + \varepsilon_{mp}k_{CP}d^4_{i-toMem}, & \text{if } \; d < d_0 \\ k_{CP}E_{elect} + \varepsilon_{fs}k_{CP}d^2_{i-toMem}, & \text{if } \; d > d_0 \end{cases} \tag{9}$$

And the members spent energy to receive the TDMA schedules from the CH is computed by Eq. (2).

4.3.2. *Steady phase*

In the steady state, active nodes transmit and sense data in the form of packets k to their CH based on the TDMA schedule received from the sink. Within a cluster, each CH is always prepared to accept this sensed data from its members. All sensed data received by the CH is aggregated and converted into a single data stream before being transmitting to the sink for processing. The CH sensor transceiver's spent energy E_{DA} to perform the aggregation task is calculated using Eq. (11). The overall energy dissipated by all members to transmit sense data to their CHs is calculated using:

$$E_{Rx(m_i)}(k) = \sum_{i=1} m_i k E_{elec} \tag{10}$$

where m_i represents the member nodes in the series $i = 1, 2, 3, \ldots, n - L$. n and L denote the total number of all sensor nodes and cluster heads respectively. The energy spent by the CH to aggregate sensed data from its members and itself is calculated using:

$$E_{DA(m_i+1)}(k) = kE_{DA} * \left(\sum_{i=1} m_i + 1 \right) \tag{11}$$

Lastly, the CH dissipates energy to send their aggregated data to the sink and this can be calculated using:

$$E_{Tx(ch_i)}(k_{CP}, d_{i-toSink}) = \sum_{i=1} ch_i * \begin{cases} k_{CP}E_{elect} + \varepsilon_{mp}k_{CP}d^4_{i-toSink}, & \text{if } \; d > d_0 \\ k_{CP}E_{elect} + \varepsilon_{fs}k_{CP}d^2_{i-toSink}, & \text{if } \; d < d_0 \end{cases} \tag{12}$$

4.3.3. *Total energy consumption*

The overall energy spent by all CHs can be calculated using:

$$\begin{aligned} E_{CHs} &= 2 * E_{Rx}(k_{CP}) + E_{Tx}(k_{CP}, d_{i-toSink}) + E_{Tx}(k_{CP}, d_{i-toMem}) \\ &\quad + E_{Rx(m_1)}(k) + E_{DA(m_i+1)}(k) \end{aligned} \tag{13}$$

where $2 * E_{Rx}(k_{CP})$ results from the fact that a CH dissipates energy twice, when it receives requests for ID, position and energy levels; and secondly when it receives membership status information for cluster set-up from the sink via a control packet. The energy lost by the member node is calculated as:

$$E_{Mem} = E_{Tx}(k_{CP}, d_{i-toSink}) + E_{Tx}(k_{CP}, d_{i-toCH}) + 3 * E_{Rx}(k_{CP}) \tag{14}$$

where $3 * E_{Rx}(k_{CP})$ expresses that energy is lost by each member node when receiving control packets. $2 * E_{Rx}(k_{CP})$ is the same as explained above and an additional loss occurs when receiving

Table 1
Parameter settings for homogeneous WSNs scenarios.

Experiment	Parameter			
	Number of sensors	Sink coordinates (m)	Deployment area (m^2)	Initial energy (J)
Exp_{R0M100}	100	(50,175)	100 × 100	$\mu = 0.5$, $\sigma_M = 0$
Exp_{R0M400}	400	(50,200)	100 × 100	$\mu = 0.5$, $\sigma_M = 0$
$Exp_{R0M1000}$	1000	(50,350)	200 × 200	$\mu = 1.0$, $\sigma_M = 0$

Table 2
Parameter settings for heterogeneous WSNs scenarios.

Experiments	Parameter				
	Number of heterogeneous nodes (R)	Number of homogeneous nodes (M)	Sink coordinates (m)	Deployment area (m^2)	Initial energy (J)
Exp_{R25M0}	25	0	(50, 175)	100 × 100	$\mu = 0.5$, $\sigma_R = 0.05$
Exp_{R50M0}	50				
Exp_{R75M0}	75				
Exp_{R100M0}	100				
Exp_{R25M75}	25	75	(50, 175)	100 × 100	$\mu = 0.5$, $\sigma_R = 0.05$, $\sigma_M = 0$
Exp_{R50M50}	50	50			
Exp_{R75M25}	75	25			

TDMA schedules from its CH. The total energy dissipated by all nodes is computed as:

$$E_{TOTAL} = E_{CHs} + E_{Mem} \tag{15}$$

Note: Current residual energy E_{Res} of each node is calculated by subtracting the total energy consumption from the residual energy of previous round.

5. Simulation results

The performance of clustering protocols can be evaluated using different types of metrics [27]. In this work, a MATLAB simulation model was developed to test the performance of our proposed algorithm in terms of lifetime evaluation of sensor nodes. The experimental conditions for all of the trials investigated are presented in Tables 1 and 2. In each simulation run, the sensors under test are randomly redistributed in an x,y grid with origin 0,0 and a deployment area of 100 m × 100 m or in the case of 1000 nodes over 200 m × 200 m. Each trial has only one sink that is placed at a location outside the sensor deployment area with coordinates provided in Tables 1 and 2; the number of CHs is dynamic.

After clustering, the estimated maximum distances between a member node and a CH were found to be 39.20 m, 29.43 m and 26.17 m for 100, 400 and 1000 sensor trials respectively. Also, the estimated maximum distance between a CH and the sink node were 126.55 m, 141.82 m and 303.42 m for the 100, 400 and 1000 sensor trials respectively. Our proposed HACH protocol is considered scalable in sense that it improves its energy efficiency as the network size increases. To demonstrate this fact we compare the performance of our proposed protocol with SEECH, TCAC and SEECH protocols using experiments Exp_{R0M100}, Exp_{R0M400}, $Exp_{R0M1000}$ which represent 100, 400 and 1000 homogeneous (same energy levels) sensor nodes respectively and zero heterogeneous (different energy levels) nodes in terms of initial energy value (refer to Table 1). Also, Table 2 presents experiment Exp_{R25M0}, Exp_{R50M0}, Exp_{R75M0}, Exp_{R100M0} which has 25, 50, 75, 100 heterogeneous sensor nodes respectively and no homogeneous nodes. Lastly, the authors conducted more experiments that mixed heterogeneous nodes with homogeneous nodes, namely experiments Exp_{R25M75}, Exp_{R50M50}, Exp_{R75M25}. The communication parameters used for all the experiments presented in Tables 1 and 2 is shown in Table 3.

In addition to the simulation parameters in Table 3, the GA parameters are set as population size, $p_s = 100$ and mutation rate, $p_m = 0.05$. R and M signify the number of heterogeneous and homogeneous sensor nodes respectively. In Tables 1 and 2, μ represents the sensor nodes mean energy, σ_R and σ_M represent the standard deviation of heterogeneous and homogeneous sensor nodes respectively. For all experiments in Table 2, the mean initial energy E_0 used is 0.5 J.

5.1. Stability period and network lifetime

The stability period length (SPL) is the time range from the start of network operation until when the first node dies (FND) whereas the instability period (IPL) is the time span from the FND until the last node dies (LND). The WSN lifetime is the time range from the start of network operation until the last node dies, which exclude energy unlimited sink devices (refer to Section 3). Immediately after the last sensor dies, the WSNs will stop its operation because the sink has lost its connectivity from the sensors. Alternatively, the WSNs lifetime can be defined as the combination of stability and the instability period. A reliable clustering process is characterized by a long SPL and a short IPL. Experimental results shown in Fig. 7 depict the number of nodes that are alive after each round.

The performance of our protocol is compared with other protocols in terms of the FND, LND, and IPL measures as seen on the graphs presented in Fig. 7. Table 4 shows that our HACH protocol maintains the network operational lifetime of 338, 131 and 36 more than the LEACH, TCAC and SEECH respectively for Experiment Exp_{R0M100}. For a medium density WSN scenario Exp_{R0M400}, our HACH shows a longer lifetime of 1235 rounds compared with LEACH, TCAC and SEECH which have a lower value of 685, 948 and 1016 respectively. The most fascinating result is that under the most dense WSNs ($Exp_{R0M1000}$) containing 1000 sensors, our algorithm gives extremely high value of 1789 rounds compared with

Table 3
Communication parameters with specified values.

Parameter	Value
Electronics energy, E_{elect}	50 nJ/bit
Multipath loss, ε_{mp}	0.0013 pJ/bit/m^4
Free space loss, ε_{fs}	10 pJ/bit/m^2
Aggregation energy, E_{DA}	5 nJ/bit/signal
Threshold distance, d_0	87 m
Control packet size, k_{CP}	50
Packets size, k	400

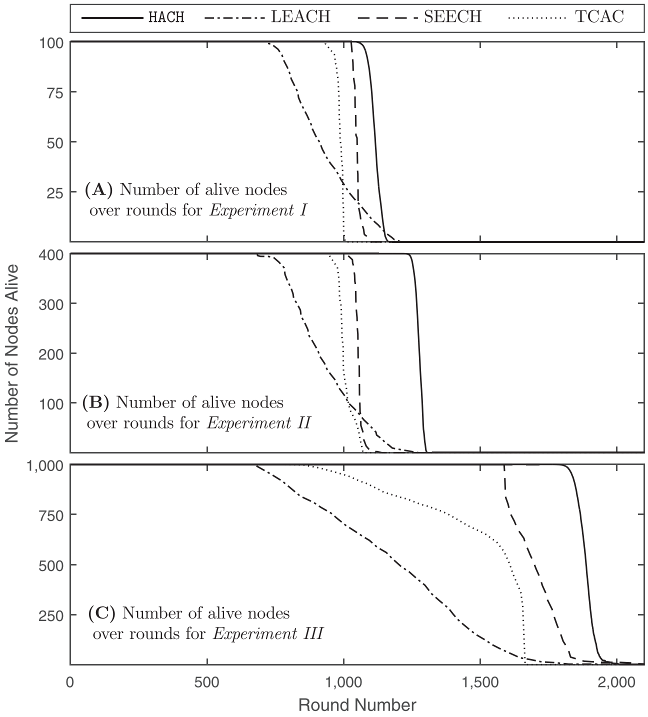


Fig. 7. Lifetime evaluation of HACH, LEACH, SEECH and TCAC.

672, 725 and 1587 round of LEACH, TCAC and SEECH respectively. This shows that as the network size increases, the performance of HACH algorithm continues to improve.

Also, for Experiments Exp_{R0M400} and $Exp_{R0M1000}$ as shown in Fig. 4, it was deduced that HACH has a very low IPL values for larger network sizes apart from Experiment Exp_{R0M100} which has 30 rounds more than the TCAC protocol. This means that HACH works very well in larger and denser network size. It is also noteworthy that the FND obtained in our proposed HACH protocol for Exp_{R25M0} (see Table 6) is 54 rounds more than LEACH protocol (refer to Exp_{R100M0} in Table 1); which means that are protocol can still perform with fewer nodes than the LEACH protocol.

5.2. Average energy at first node dies (AEFND)

The AEFND is defined as the sum of all current or residual energy values of the sensor nodes divided by the number of nodes at the round when the first node dies. Many nodes begin to die when the first node dies and during the instability periods because of the depleted energy supply. In the HACH protocol, energies of some nodes are balance until the FND time and this is indicated on the graphs of Fig. 7 by a sharp decline in the number of nodes that are alive for HACH, SEECH and TCAC protocol. One of the performance goals for an energy efficient protocol is to keep the AEFND to a very low value and our HACH protocol kept the AEFND to a very low value of approximately zero for all experiments as shown in Tables 5 and 6. For example, Experiment Exp_{R0M100} has an AEFND of 0.0232 J at FND time of 1064 as shown in Fig. 8.

This proves the fact that we were able to manage the energy usage until the FND time. The low AEFND values in Table 6 means that our protocol can efficiently manage energy consumption under heterogeneous WSN environments. Therefore, our proposed HACH reduces the energy consumed and enhances energy balance across the nodes in the sensor field thereby extending the network lifespan.

Table 4
Performance comparison of LEACH, TCAC and SEECH with HACH.

Experiment	Protocol	Performance measure (round)		
		FND	LND	IPL
Exp_{R0M100} (100 nodes)	LEACH	726	1209	483
	TCAC	933	1006	73
	SEECH	1028	1099	71
	HACH	1064	1167	103
Exp_{R0M400} (400 nodes)	LEACH	685	1274	589
	TCAC	948	1071	123
	SEECH	1016	1140	124
	HACH	1235	1307	72
$Exp_{R0M1000}$ (1000 nodes)	LEACH	672	2014	1342
	TCAC	725	1664	939
	SEECH	1587	2202	615
	HACH	1789	2010	221

Table 5
AEFND of proposed HACH protocol.

	Experiments		
	Exp_{R0M100}	Exp_{R0M400}	$Exp_{R0M1000}$
AEFND	0.0232	0.0164	0.0650

Table 6
Performance measures for different heterogeneous WSN scenarios.

Experiment	Performance measures			
	FND	LND	IPL	AEFND
Exp_{R25M0}	780	937	157	0.040608
Exp_{R25M75}	975	1126	151	0.033479
Exp_{R50M0}	863	1010	147	0.033479
Exp_{R50M50}	976	1061	147	0.030858
Exp_{R75M0}	920	1059	139	0.033468
Exp_{R75M25}	972	1123	151	0.030196
Exp_{R100M0}	971	1110	139	0.033168

5.3. WSNs heterogeneity

After a certain number of rounds when the sensor networks lifetime has been depleted, new nodes are introduced to re-energize the sensor network. These new nodes are equipped with a higher constant energy value and nodes that are already in use have lower random energy, resulting in energy heterogeneity [31]. As shown in Fig. 9, the FND value decreases from 1064 for Exp_{R0M100} (refer to Table 4) to FND of 780 in Exp_{R25M0} (refer to Table 6). Despite the increase in the ratio value of heterogeneous to homogeneous sensors from 25 to 100, which introduces more complexities in terms of energy imbalance, our protocol was still able to balance the energy consumption and maintain a constant FND value.

This phenomenon of starting a network operation with unbalanced energy distribution in a sensor networks is called WSNs heterogeneity. In this paper, the experiments that falls under the three level of energy heterogeneity are as follows:

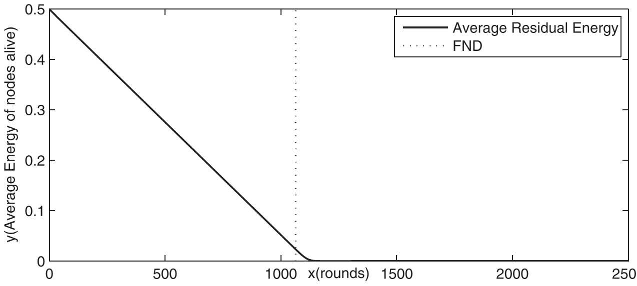


Fig. 8. Average residual energy of nodes alive versus rounds (refer to Exp_{R0M100}).

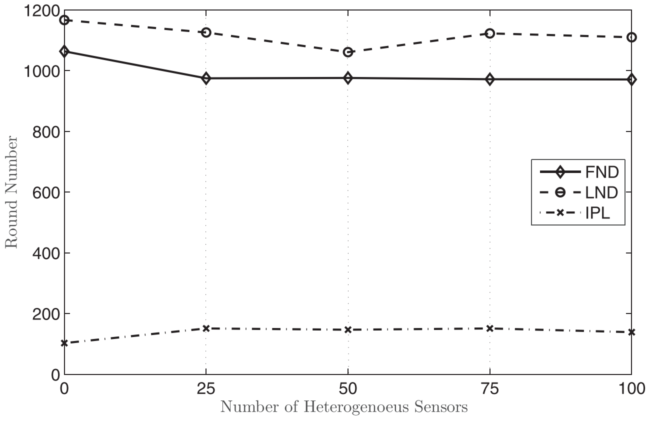


Fig. 9. Round number versus numbers of heterogeneous sensors.

- One-Quarter Level: Experiment Exp_{R25M0} and Exp_{R25M75}.
- Half Level: Experiment Exp_{R50M0} and Exp_{R50M50}.
- Three-Quarter Level: Experiment Exp_{R75M0} and Exp_{R75M25}.

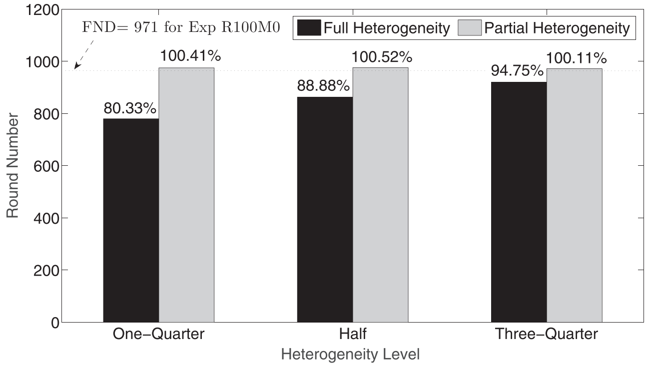
Each level has experiments with Full and Partial heterogeneity. Also, it can be observed in Table 6 that adding some energy-homogeneous sensor nodes to a set of energy-heterogeneous or energy depleted sensors extends the lifetime by a considerable amount, for example experiments Exp_{R25M75}, Exp_{R50M50} and Exp_{R75M25} has a FND round of 195, 113 and 52 greater than experiments Exp_{R25M0}, Exp_{R50M0} and Exp_{R75M0} respectively. The performance of each experiment is compared with Exp_{R100M0}, and their percentage value is shown on top of each bar as shown in Fig. 10.

5.3.1. *Full heterogeneity*

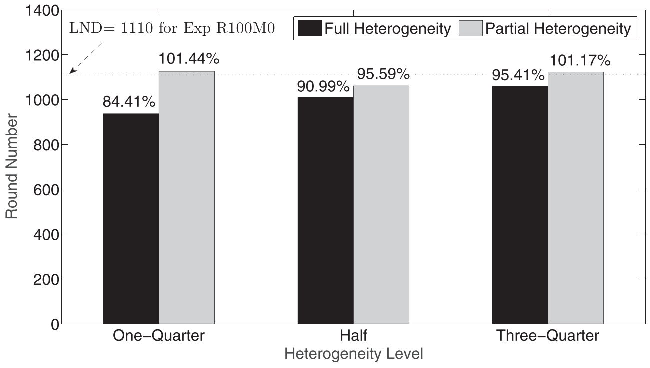
Full heterogeneity refers to a scenario whereby all the sensor nodes in a sensing field have random energy values and zero number of constant energy value. For example in Table 2, experiments Exp_{R25M0}, Exp_{R50M0}, Exp_{R75M0} and Exp_{R100M0} are conducted using 25, 50, 75 and 100 number of sensor nodes with random energy values and 0 constant energy values for all the experiments. The bar charts presented in Fig. 10 show that performance improves from one-quarter to the three-quarter full heterogeneity level when compared with Exp_{R100M0}. In Fig. 10a, FND percentages of increasing order of 80.33%, 84.41% and 94.75% were obtained. Also, the LND percentage is in ascending order of 84.41%, 90.99%, 95.41% as shown in Fig. 10b. Additionally the IPL percentage is in decreasing order of 112.95%, 105.76%, 100.0%; meaning the performance increased as the number of heterogeneous nodes increased. Also, in Fig. 10c, Exp_{R50M0} was able to obtain 105.76% which is the same value as the half-level Exp_{R50M50}.

5.3.2. *Partial heterogeneity*

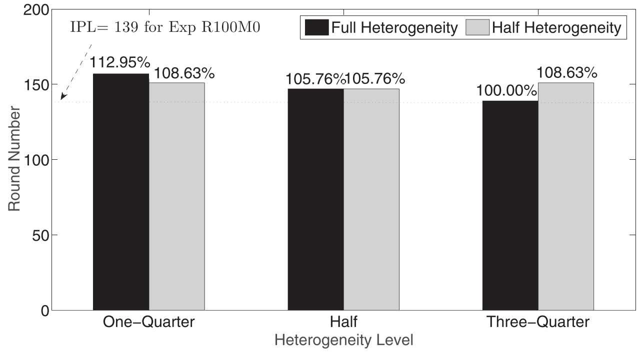
This is the WSN scenario that describes the ratio combination of sensor nodes with random and constant energy values. In Table 6, Exp_{R25M75}, Exp_{R50M50} and Exp_{R75M25} use 25, 50, 75 sensor nodes with random energy and 75, 50, 25 sensor nodes with constant energy respectively. In Fig. 10a, the FND time for Exp_{R25M75}, Exp_{R50M50}, and Exp_{R75M25} is 100.41%, 100.52% and 100.11% respectively when compared with Exp_{R100M0}; showing that there is no significant improvement as the ratio of heterogeneous to homogeneous nodes increases. In Fig. 10, Exp_{R50M50} produces the most improved FND of 0.52% more than the Exp_{R100M0} and percentage reduction of LND by 4.41%.



(a) FND



(b) LND



(c) IPL

Fig. 10. Performance comparison of different WSNs heterogeneity level for (a) FND, (b) LND and (c) IPL measures.

6. Conclusion

In this paper, we have proposed a new `HACH` algorithm. The algorithm reduces and balances energy consumption by selecting distributed nodes with high energy as cluster heads to prolong network lifetime. Sequentially, this is achieved by two major operations such as sleep scheduling and cluster head selection operations. The `SSIN` sleep scheduling mechanism inspired by Boltzmann selection process was proposed to decide which nodes to send into sleep mode with negligible effect on the coverage. Subsequently, we employed a genetic algorithm-based technique called the `HEECHS` protocol that would distribute cluster heads evenly within a sensor field to ensure that energy consumption is balanced across the networks. To guarantee an efficient cluster head selection process, we designed an objective function to evaluate the quality of our solutions. Simulation results of the first three experiments shows that our proposed HACH algorithm outperforms the SEECH, TCAC and LEACH. Also, further experiments demonstrated that our protocols can perform even better under different heterogeneity levels of wireless sensor network settings and still maintain acceptable performances.

Acknowledgement

This work was supported by the UK Innovate/EPSRC grant EP/M506734/1.

References

[1] M.O. Oladimeji, M. Turkey, S. Dudley, A heuristic crossover enhanced evolutionary algorithm for clustering wireless sensor network, in: Applications of Evolutionary Computation – 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings, Part I, 2016, pp. 251–266.

[2] S. Naeimi, H. Ghafghazi, C.-O. Chow, H. Ishii, A survey on the taxonomy of cluster-based routing protocols for homogeneous wireless sensor networks, Sensors 12 (6) (2012) 7350–7409.

[3] A. Chakraborty, S.K. Mitra, M.K. Naskar, Energy efficient routing in wireless sensor networks: a genetic approach, CoRR abs/1105.2090.

[4] A.A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks, Comput. Commun. 30 (14) (2007) 2826–2841.

[5] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wireless Commun. 1 (4) (2002) 660–670.

[6] W.E. Hart, N. Krasnogor, J.E. Smith, Recent Advances in Memetic Algorithms, vol. 166, Springer Science & Business Media, 2005.

[7] C. Danratchadakorn, C. Pornavalai, Coverage maximization with sleep scheduling for wireless sensor networks, in: IEEE 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015, pp. 1–6.

[8] B. Singh, D. Lobiyal, Energy preserving sleep scheduling for cluster-based wireless sensor networks, in: IEEE Sixth International Conference on Contemporary Computing (IC3), 2013, 2013.

[9] E. Bulut, I. Korpeoglu, Sleep scheduling with expected common coverage in wireless sensor networks, Wireless Netw. 17 (1) (2011) 19–40.

[10] S.H. Kang, T. Nguyen, Distance based thresholds for cluster head selection in wireless sensor networks, IEEE Commun. Lett. 16 (9) (2012) 1396–1399.

[11] M. Ye, C. Li, G. Chen, J. Wu, Eecs: an energy efficient clustering scheme in wireless sensor networks, in: Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International, IEEE, 2005, pp. 535–540.

[12] N. Dimokas, D. Katsaros, Y. Manolopoulos, Energy-efficient distributed clustering in wireless sensor networks, J. Parallel Distrib. Comput. 70 (4) (2010) 371–383.

[13] S. Lin, J. Zhang, G. Zhou, L. Gu, J.A. Stankovic, T. He, ATPC: adaptive transmission power control for wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, 2006, pp. 223–236.

[14] V. Loscri, G. Morabito, S. Marano, A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH), in: IEEE Vehicular Technology Conference, vol. 62, IEEE; 1999, 2005, p. 1809.

[15] O. Younis, S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans. Mobile Comput. 3 (4) (2004) 366–379.

[16] D.P. Dahnil, Y.P. Singh, C.K. Ho, Topology-controlled adaptive clustering for uniformity and increased lifetime in wireless sensor networks, IET Wireless Sens. Syst. 2 (4) (2012) 318–327.

[17] M. Tarhani, Y.S. Kavian, S. Siavoshi, SEECH: Scalable energy efficient clustering hierarchy protocol in wireless sensor networks, IEEE Sensors J. 14 (11) (2014) 3944–3954.

[18] S. Bayrakli, S.Z. Erdogan, Genetic algorithm based energy efficient clusters (GABEEC) in wireless sensor networks, Procedia Comput. Sci. 10 (2012) 247–254.

[19] N. Latiff, C.C. Tsimenidis, B.S. Sharif, Energy-aware clustering for wireless sensor networks using particle swarm optimization, in: IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007, IEEE, 2007, pp. 1–5.

[20] J.-L. Liu, C.V. Ravishankar, et al., LEACH-GA: Genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks, Int. J. Machine Learn. Comput. 1 (1) (2011) 79–85.

[21] K.G. Vijayvargiya, V. Shrivastava, An amend implementation on leach protocol based on energy hierarchy, Int. J. Curr. Eng. Technol. 2 (4) (2012) 427–431.

[22] W. Charfi, M. Masmoudi, F. Derbel, A layered model for wireless sensor networks, in: IEEE 6th International Multi-Conference on Systems, Signals and Devices, 2009. SSD'09, 2009, pp. 1–5.

[23] D. Dumitrescu, B. Lazzerini, L. Jain, A. Dumitrescu, Evolutionary Computation, International Series on Computational Intelligence, Taylor & Francis, 2000.

[24] D.E. Goldberg, A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing, Complex Syst. 4 (4) (1990) 445–460.

[25] Q. Mamun, A qualitative comparison of different logical topologies for wireless sensor networks, Sensors 12 (11) (2012) 14887–14913.

[26] V. Pal, G. Singh, R. Yadav, Analyzing the effect of variable round time for clustering approach in wireless sensor networks, Lect. Notes Softw. Eng. 1 (1) (2013) 31.

[27] T. Lixin, Improved genetic algorithms for TSP, J. Northeastern Univ. (Natural Science) (1999) 01.

[28] B.S. Hasan, M. Khamees, A.S.H. Mahmoud, et al., A heuristic genetic algorithm for the single source shortest path problem, in: IEEE/ACS International Conference on Computer Systems and Applications, 2007. AICCSA'07, 2007, pp. 187–194.

[29] R. Halke, V. Kulkarni, En – LEACH routing protocol for wireless sensor network, Int. J. Eng. Res. Appl. 2 (4) (2012) 2099–2102.

[30] J. Brunda, B. Manjunath, B. Savitha, P. Ullas, Energy aware threshold based efficient clustering (EATEC) for wireless sensor networks, Energy 2 (4) (2012).

[31] H. Kour, A.K. Sharma, Hybrid energy efficient distributed protocol for heterogeneous wireless sensor network, Int. J. Comput. Appl. 4 (6) (2010) 1–5.