CrossMark

An information entropy based-clustering algorithm for heterogeneous wireless sensor networks

Walid Osamy[1,4] · Ahmed Salim[3,4] · Ahmed M. Khedr[2,3]

Abstract

This paper proposes a novel dynamic, distributive, and self-organizing entropy based clustering scheme that benefits from the local information of sensor nodes measured in terms of entropy and use that as criteria for cluster head election and cluster formation. It divides the WSN into two-levels of hierarchy and three-levels of energy heterogeneity of sensor nodes. The simulation results reveal that the proposed approach outperforms existing baseline algorithms in terms of energy consumption, stability period, and the network lifetime.

Keywords Clustering · Entropy coefficient · Network lifetime · Self-organizing · Three-level energy heterogeneity · Wireless sensor network

1 Introduction

A wireless sensor network (WSN) consists of spatially distributed autonomous devices known as sensors. Each senor is capable of performing some basic processing, gathering sensory information and communicating with other connected sensor nodes in the network [1, 2].

Cluster topology is a WSN topology architecture where sensor nodes are grouped into various sets called clusters. Each cluster has a cluster head (CH) with one or more cluster member (CM) sensor nodes. This topology uses multihop routing to transmit data from the source sensors to the base station (BS). Clustering approach reduces communication overheads and due to effective allocations of resource as a result, decrease the overall energy consumption and reduce the interference among sensor nodes [3]. Clustering approach reduces the size of collected data by keeping only significant information by applying data aggregation techniques at CHs [4]. The aggregated data from CHs will be transmitted to the BS. Figure 1 shows the functionality of a cluster based WSN.

Information entropy theory such as administration entropy, environment entropy, and economy entropy [5] have been employed in many discrete areas. Entropy in information theory uses the discrete probability distribution to represent the amount of uncertainty. Let X be a discrete random variable with alphabet X and probability mass function $p(x) = Pr\{X = x\}, x \in X$. The entropy $H(X)$ of a discrete random variable X is defined by [6] as follows:

$$H(X) = - \sum_{x \in X} p(x) log_2 p(x) \tag{1}$$

The minimum entropy is 0 and it occurs when one of the probabilities is 1 and the rest are 0s, while the entropy achieves the maximum value $(H_{max} = log_2(n))$ when all the probabilities have equal values of $\frac{1}{n}$, where n is the number of outcomes.

Entropy indicates the valuable information produced by the data, as a result it can be used to determine the weights. The entropy value becomes smaller when the analyzed objects have fairly big difference between each others on a

✉ Walid Osamy
walid.osamy@fci.bu.edu.eg; w.Elsherif@qu.edu.sa

Ahmed Salim
a.salem@qu.edu.sa

[1] Department of Computer Science, Faculty of Computers and Informatics, University of Benha, Benha, Egypt

[2] Department of Computer Science, College of Sciences, University of Sharjah, Sharjah, United Arab Emirates

[3] Mathematics Department, Faculty of Science, Zagazig University, Zagazig, Egypt

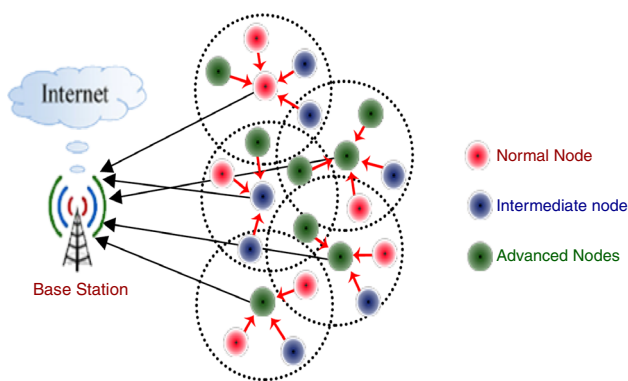[4] Qassim University, P.O. Box 931, Buridah, Kingdom of Saudi Arabia

Fig. 1 A heterogeneous WSN model with three types of sensor nodes

given specific criterion. However, the weight of the criterion should be smaller when the difference between objects is smaller (larger entropy value). Consequently, the entropy coefficient method is a target empowering method to determine the weight by calculating the entropy weights of each criterion based on determining variation degree with respect to every criterion value [5, 7–9].

Using the local information of the sensor nodes such as residual energy, a clustering algorithm can be adopted in terms of entropy as the election criterion. Such a new algorithm is energy efficient, moreover, we consider selection of CH node as a decision problem based on multiple criteria such as residual energy, sensor node density, and distance from the BS. Hence, we have multiple alternatives (set of sensor nodes) where each alternative consists of multiple criteria (i.e., sensor node's information) and we need to make selection decision. For that, one of multi-criteria decision analysis (MCDA) methods is used to solve the decision problem with multiple criteria. In our proposed algorithm, the weighted product model (WPM) [10] is utilized for addressing the decision problem and the entropy weight coefficient [5, 7, 9, 11] is used to assess the weight of different criteria. Our simulation results reveal that the proposed strategy can manage power consumption better than existing algorithms and achieves the desired results for WSNs. The main contribution of this paper can be summarized as follows:

1. A new dynamic, distributive, self-organizing, and energy-efficient clustering algorithm that uses sensor nodes' local information (e.g., residual energy) in terms of entropy as the election criterion is proposed. The new algorithm does the following:

   - Organizes the WSN into two-levels of the hierarchical network and consider three-levels of energy heterogeneity of sensor nodes.
   - Considers the selection of CH as a decision problem with multiple criteria such as residual energy, node density, and distance from the BS.

2. We provide a solution to the decision problem through Weighted Product Model (WPM) where sensor nodes represent the multiple alternatives and each alternative consists of multiple criteria (sensor nodes information).

3. We introduce the concept of entropy weight coefficient to assess the weight of different criteria which reflects the current status of node information in the network.

4. Provide extensive simulation results to prove that the proposed strategy can manage power consumption and substantially outperforms the baseline algorithms in terms of energy consumption, stability period, and the network lifetime.

The remainder of this paper is structured as follows: the discussion of the related research is given in Sect. 2. The considered models description is presented in Sect. 3. Section 4 describes the proposed algorithm. In Sect. 5, the performance analysis and results are presented. Evaluations indicate that the proposed approach efficiently solves the

Table 1 Table of notations

Notation	Description
CH	Cluster head
CM	Cluster member
WPM	Weighted product model
EWC	Entropy weight coefficient method
e_{CH}	Total energy consumption by CH
$E_{consumed}(s, r)$	Total energy consumption by node s at round r
ψ	List of nominated nodes to be CHs
E_0	Initial energy of normal node
E_{int}	Initial energy for intermediate node
E_{adv}	Initial energy for advanced node
E_{Rx}	Energy consumed in reception of data
E_{Tx}	Energy consumed in transmission of data
ϵ_{mp}	Amplification energy expended to overcome the multi-path
ϵ_{fs}	Amplification energy expended to overcome the free space
E_{elec}	Electronics energy
D	Distance to BS
E	Residual energy
DN	Sum of distances to neighbor nodes
$N_{support}$	Number of nodes that can be supported
n	Total number of sensors
m	Ratio of advanced sensor nodes
k	Number of clusters
b	Ratio of intermediate sensor nodes
A_L	alternative number L
a_{ij}	Performance value of the alternative A_i and criteria number j
H_i	Entropy of criteria i
w_i	Entropy coefficient weight of criterion i

problem and exceeds other existing algorithms. The conclusion of the proposed work is given in Sect. 6. Used notations through the paper are given in Table 1.

2 Related research

Several recent works address the problem of clustering in a WSN. LEACH [12], was the first most routing protocol that used clustering concept in a WSN. LEACH collects data in rounds. Each round is split into two main phases: setup phase where a specific ratio of sensor nodes are selected as CHs according to their probability in particular round. Each non-CH joins the nearest CH as a cluster member (CM). Then in the second phase, each CH determines a schedule for its CM to transmit its data without any collision. In the steady state phase, each CM transmits its data in assigned time slot.

LEACH has many variant versions such as [3, 13–27]. In [13], a centralized cluster formation algorithm (LEACH-C) is designed such that there are a certain number of clusters, k, during each round. In [14], to enhance LEACH-C and LEACH, the authors proposed a technique to optimize LEACH (O-LEACH) by dynamically selecting CH according to the residual energy of sensor nodes.

Deterministic energy efficient clustering protocol (DEC) is presented in [16]. DEC is three-level energy sensor nodes and autonomously select CHs according to their energy levels. Only the set-up phase in LEACH is modified and the BS takes part in the election of the CHs in the first round. As in LEACH, when the steady phase begins, each sensor node transmits its data to the nearest CH that performs data aggregation. However, in DEC, before the end of steady phase, each CH checks the piggybacked of received cluster member data to decide whether it will remain CH or will relinquish its role by choosing another sensor node among its members with the highest energy as the new CH. In the next round, the new selected CHs broadcast their role then the non CHs will select their CH and the steady phase begins again. This process continues in each round until the last sensor node dies. In [19], the authors proposed intra-balanced LEACH protocol to extend LEACH by balancing the energy consumption in WSN. In [27], the authors proposed a protocol that improves the CH election, the special node processing and inter cluster routing problem, and then an improved protocol called LEACH-Impt was proposed.

In [21], the authors proposed LEACH with two level CH (LEACH-TLCH) protocol that deploys a secondary CH (2CH) to relieve the CH burden in these circumstances. However, in LEACH-TLCH the optimal distance of CH to BS, and the choicest CH energy level for the 2CH to be deployed for achieving an optimal network lifetime was not considered. In [22], the authors improved LEACH-TLCH by investigating the conditions set to deploy the 2CH for an optimal network lifetime. In [23], a hybrid unequal energy efficient clustering is proposed to increase lifetime of the network. In the proposed protocol, a new mechanism based on arrangement of nodes in a network to determine whether nodes should use information of their neighbors or should not use this information.

In [24], a protocol in form of cluster-head restricted energy efficient protocol (CREEP) has been proposed to overcome high system complexity due to computation and selection of large number of CHs and improve the network lifetime by modifying the CH selection thresholds in a two level heterogeneous WSN. The work in [25] focuses on reducing the energy dissipation thereby improving the network lifetime of the randomly deployed senor nodes in a WSN and also improving the rate of data packets forwarded to the BS at the same time. This is achieved by dividing the deployment region into various sectors and then selecting the CHs from each sector based on highest remaining energy of the sensors. The work in [26] reviews the routing concepts for diverse heterogeneous WSNs scenarios and covers the state of the art in the area. It also discusses the effects and inter-dependencies of different heterogeneities in routing decisions and unveils new research directions in the area.

The authors in [17] presented stability period protocol (SEP) which is based on weighted election probabilities using the remaining energy of each sensor node to be selected as CH. The energy heterogeneity problem in the network is characterized using two-node power classification (nodes are labeled with "normal" and "advanced"). A variant of LEACH called extended heterogeneous LEACH (EHE-LEACH) is proposed in [18]. EHE-LEACH uses same model as in SEP. M-SEP [28] is a modified version of SEP. M-SEP enhances the network lifetime by considering the energy of nodes in a current ongoing round and an average energy of the whole network.

In [29], the authors proposed distance incorporated modified stable election protocol (D-MSEP) which is an improved version of modified stable election protocol (M-SEP) as it incorporates the distance factor in probabilistic formula of CH selection in each cluster. The inclusion of distance factor helps in avoidance of CH selection farther from the BS. This leads to saving of huge amount of energy leading to the escalated stability period.

A heterogeneity-aware energy efficient clustering (HEC) algorithm is suggested in [20]. HEC selects CHs based on the three network lifetime phases: only advanced sensor nodes are allowed to become CHs in the initial phase; in the second active phase, all the sensor nodes are allowed to participate in CH selection process with equal probability, and in the last dying out phase, clustering is relaxed by allowing direct transmission. A distributed energy-efficient clustering scheme for heterogeneous WSNs (DEEC) is presented in [30]. DEEC follows the

same rules of energy-heterogeneity of SEP. An Enhanced Distributed Energy Efficient Clustering scheme (E-DEEC) is proposed in [31] by expanding the DEEC to three-level heterogeneity. However, CHs selection probabilities are not adapted as per the sensor nodes energy levels [32].

A three level heterogeneous network model for WSNs is proposed in [33]. The proposed heterogeneous WSN model selects CHs and their respective CMs by using weighted election probability and a threshold function. The proposed model is implemented based on DEEC. An enhanced developed distributed energy-efficient clustering (EDDEEC) with heterogeneous network model is proposed in [32, 34] that is established on three energy levels of sensor nodes. EDDEEC uses clustering mechanism which dynamically modifies the CH election probability. Modified enhanced developed distributed energy-efficient clustering (MED-DEEC) algorithm has been presented in [35]. MED-DEEC employs an energy-aware protocol similarity in [32, 34]. However, the decision of electing CHs in MED-DEEC is based on the remaining and initial energy of the sensor node, average distance and average energy of the network. A distributed unequal cluster-based routing (DUCR) proposed in [36] is a distributed multi-objective-based clustering algorithm. DUCR uses multi-objective optimization technique to assign CMs to an appropriate CH so that the load is balanced. The work in [37] proposed a distributed energy-efficient clustering protocol (DCE) for heterogeneous WSNs. DCE is based on a Double-phase CH Election scheme where, the CH election process is divided into two phases. In the first phase, tentative CHs are selected based on initial and residual energy of sensor nodes. In the second phase, tentative CHs are replaced by their CMs to form the final set of CHs if any CM in their cluster has more residual energy.

All previous works consider special attributes of sensor nodes, e.g., remaining energy, distance to base station, … etc. as criteria for selecting CHs, ignoring the cluster load, i.e., the number of cluster members that can be served by CH or the number of sensor nodes that can be supported by the CH. In our proposed work, we consider cluster load as a criterion for selecting CHs combined with the remaining energy, distance to base station and degree of neighboring sensor nodes. Moreover, during the operation of the network the CH could not have enough energy to complete its task for that we introduce a new method to predict the remaining energy of each sensor node at the end of the next round before the round starts and selects the most appropriate sensor node that can continue its work till the end of the round to be the CH. Also, in the proposed algorithm, three-level energy heterogeneity of sensor nodes are used as in [16, 31, 32, 38] and entropy weight coefficient as in [5, 7, 9, 11] is used in order to come up with optimal CHs.

Several works [5, 7–9, 11] use entropy method to take a decision. Entropy weighted multi criteria routing (EMCR)

[11] is a routing scheme in which the next hop decision is based on MCDA method along with entropy method of information theory. The work in [7] uses fuzzy comprehensive evaluation based entropy weight coefficient to analyze the intrusion detection systems (IDSs). Also, entropy weight coefficient method is used in [9] for soil quality assessment. In [8], optimization for objective weights and subjective weights is proposed based on the fuzzy multiple attributes decision making routing (MADMR) algorithm that selects next hop in a WSN. The entropy coefficient model is used in [8] to avoid excessive deviation from the objective weights. In [5], an index system for capacity assessment is developed by employing the entropy weight coefficient method.

3 Models description

In this section, the different used models in our proposed algorithm such as energy model, radio energy dissipation model, network model are described.

3.1 Energy model

Energy heterogeneity is the most significant characteristic of a WSN which includes: computational heterogeneity, link heterogeneity, and energy heterogeneity. In this paper, we address three heterogeneity energy levels as in, DEC and EDDEEC where, we have three types of sensor nodes: *normal*, *intermediate*, and *advanced*. The initial energy of intermediate is between the initial energy of normal and advanced. If E_0 is the initial energy of normal sensor nodes, then the initial energy of intermediate sensor nodes will be $E_{int} = (1 + \mu)E_0$ and of advanced sensor nodes will be $E_{adv} = (1 + \alpha)E_0$, here α and μ are the energy weight factor of advanced and intermediate sensor nodes respectively. Therefore, the total initial energy will be:

$$E_{total} = nE_0(1 + m\alpha + b\mu). \tag{2}$$

Here, n is the total number of sensors in the network, m is the ratio of advanced sensor nodes, and b represents the ratio of intermediate sensor nodes.

3.2 Radio energy dissipation model

Receiving and sending messages are the two main operations that consume energy in a WSN. The consumed energy of sending a message is greater than the consumed energy of receiving a message due to the additional energy that is required to expand the signal between the distance and the destination. Here, we use the same models (radio, data aggregation, and energy parameters) as in many previous works [12, 16, 17, 30, 32, 38]. As per previous

works, the consumed radio power to transmit *l-bits* message to a distance d is given by:

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2 & if \quad d < d_0, \\ lE_{elec} + l\epsilon_{mp}d^4 & otherwise. \end{cases} \quad (3)$$

and the consumed radio power to receive this message is:

$$E_{Rx}(l) = lE_{elec} \quad (4)$$

Here, the electronics energy consumed is E_{elec} and consumed energy by the transmitter amplifier for distance greater than d is ϵ_{mp} and for distance less than d is ϵ_{fs}, the consumed energy for aggregating data is E_{DA}, and the threshold distance is given as $d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$.

3.3 Network model

We assume that n sensor nodes are randomly deployed in a field R of size $M \times M$, the sensor nodes are static and each sensor node has a unique ID. It is also assumed that each sensor node knows its communicating neighbors, including their identifications and coordinates, which can be gathered statically via *hello* message, or periodically if frequent changes occur in the topology.

4 Entropy based clustering algorithm (EBCS)

In this section, we discuss our proposed entropy based clustering algorithm (EBCS). The main goal of EBCS is in designing a dynamic distributed and self-organizing clustering algorithm that prolongs the lifetime of the network and minimizes the consumed energy in the network.

The EBCS includes three phases: cluster heads election, cluster formation, and data transmission. A flow-graph of the proposed algorithm is shown in Fig. 2, and the detailed flow-graph is shown in Fig. 3. In cluster heads election phase, weighted product model (WPM) along with entropy weighted coefficient method (EWC) are used for resolving the election decision problem based on different criteria. In cluster formation phase, each non-CH node selects the CH that uses minimum communication cost, based on the signal strength of received advertisement. In data transmission phase, the data will be collected by CH and will be sent to BS.

The Pseudocodes that will be executed by BS, CH, and non-CH nodes are presented respectively in Algorithms 2, 3, and 4.

4.1 Cluster heads election phase

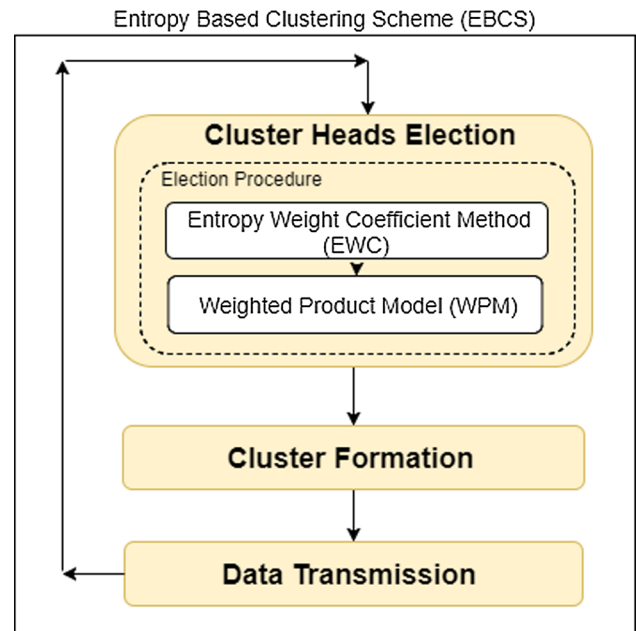WPM along with EWC are used for resolving the election decision problem based on different criteria: residual

Entropy Based Clustering Scheme (EBCS)

Fig. 2 Diagram of EBCS algorithm

energy $(E(s_j))$, number of supported sensor nodes $(N_{support}(s_j))$, distance to BS $(D(s_j))$, and sum of distance to all neighbor sensor nodes s_js $(DN(s_j))$. These criteria can be described as follows:

- *Residual energy* (E) Residual energy is the most important feature for every sensor node and the network lifetime mainly depends on the residual energy among the sensor nodes.
- *Distance to BS* (D) Distance to BS is important to be considered as more distance to BS implies more energy will be consumed in transmitting a packet.
- *Sum of distances to neighbor sensor nodes* (DN) As the CH supports its members, therefore, the number of packets transmitted over long distance between members and CHs may deplete the battery power relatively faster.
- *Number of sensor nodes that can be supported* $(N_{support})$ Since there are three-level energy heterogeneity of sensor nodes, we may have three types of CHs based on the initial energy used, CH_0, CH_{int}, and CH_{adv} with initial energy E_0, E_{int}, and E_{adv}, respectively. The energy consumption percentage of each CH type will be $c_0 = e_{CH}/E_0$, $c_{int} = e_{CH}/E_{int}$, and $c_{adv} = e_{CH}/E_{adv}$ for CH_0, CH_{int}, and CH_{adv}, respectively, where e_{CH} is given by Equation 11. It is clear that $c_0 > c_{int} > c_{adv} as E_{adv} > E_{int} > E_0$, which will result in uneven energy consumption among these CHs. Therefore, it is required to determine the number of sensor nodes that can be assisted by each type of CHs such that $c_0 \cong c_{int} \cong c_{adv}$.
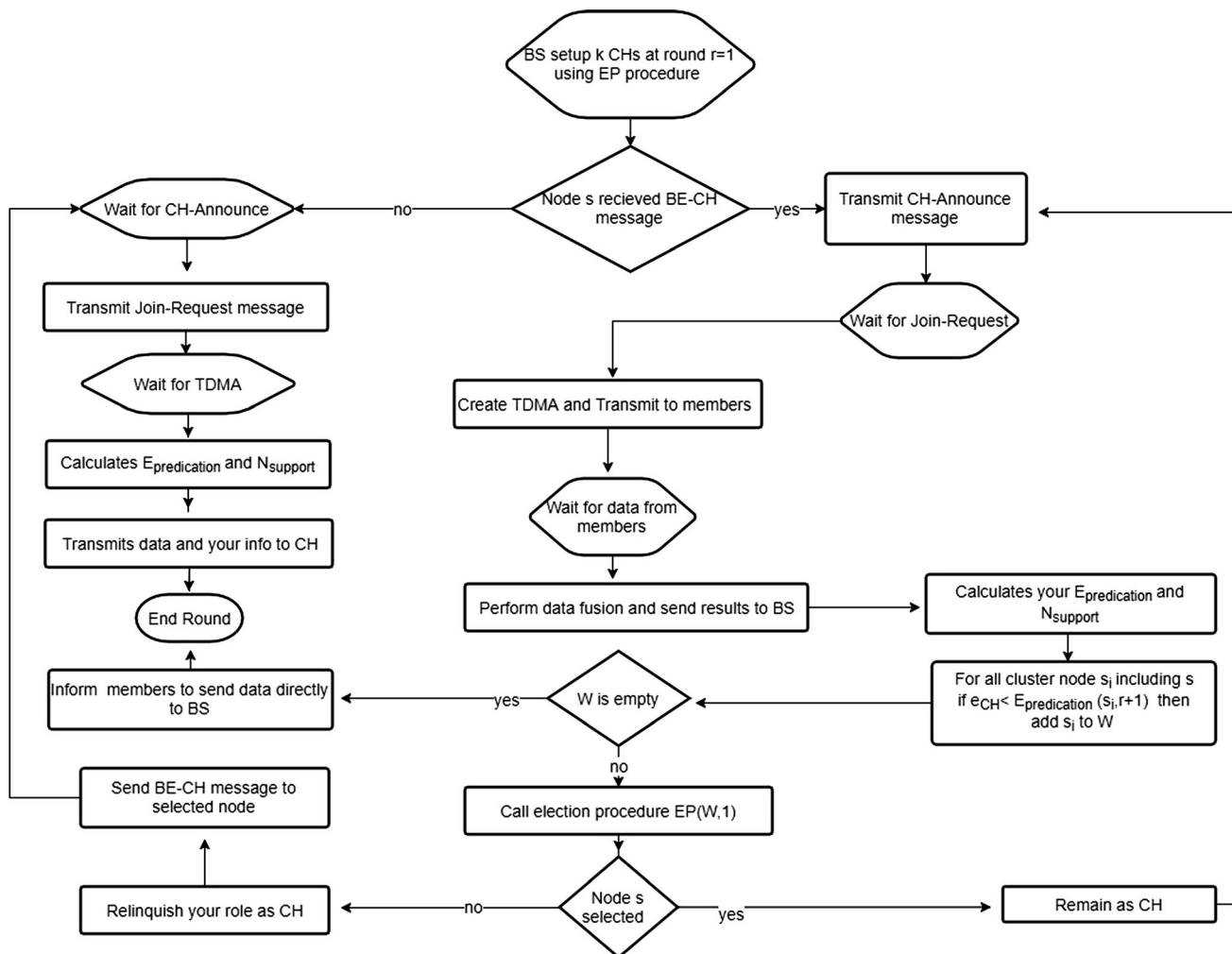
Fig. 3 Flow-graph of the proposed algorithm

The number of sensor nodes that can be assisted by each sensor node $N_{support}$ can be estimated as follows:

$$N_{support} = \left\lceil \frac{E_{remaining} * (n - k)}{\frac{k}{3}(E_0 + E_{int} + E_{adv})} \right\rceil. \tag{5}$$

Based on sensor nodes local information a decision has to be taken to select the CHs. This type of information is considered as the criteria for the election decision making (e.g., multi-criteria decision-making problem). In our proposed approach, we adapt the weighted product model (WPM) which is one of the Multi-criteria decision analysis method (MCDA) [10] and then we use entropy weight coefficient method to elect the CHs.

4.1.1 Weighted product model (WPM)

In WPM, given is a finite set of decision alternatives $(A_1, A_2, \ldots, A_n)$ described in terms of a number of decision criteria. Each decision alternative is compared with the others by multiplying a number of ratios, one for each decision criterion. Each ratio is raised to the power equivalent of the relative weight (w) of the corresponding criterion. In order to compare two alternatives A_K and A_L using WPM; the following product (P) has to be calculated for m number of criteria and n number of alternatives:

$$P\left(\frac{A_K}{A_L}\right) = \prod_{j=1}^{m} \left(\frac{a_{Kj}}{a_{Lj}}\right)^{w_j}. \tag{6}$$

Here, $K \neq L, K, L = 1, 2, \ldots, n$ and a_{ij} is the performance value of the alternative A_i. If the ratio $P(\frac{A_K}{A_L}) \geq 1$, then it implies that the alternative A_K is more useful than the alternative A_L. The best alternative is the one that is better than or at least equal to all others.

The alternative approach of WPM is the decision maker where we use only products without ratios as follows [10]:

$$P(A_K) = \prod_{j=1}^{m} (a_{Kj})^{w_j}. \tag{7}$$

In this formula, the term $P(A_K)$ indicates the performance value of alternative A_K when all the criteria are evaluated under the WPM model. The weights w_j of different criteria are calculated using entropy coefficient method that will be described in the next subsection.

4.1.2 Entropy weight coefficient method (EWC)

The entropy coefficient method is applied to determine the weights for the criteria. The main steps of calculating the weights of m criteria and n alternatives are as follows:

- Calculate the entropy of each criterion $i = 1, \ldots, m$ that can be derived from Eq. 1.

$$H_i = -\frac{1}{log_2 n} \sum_{j}^{n} p_{ij} log_2 p_{ij} \qquad (8)$$

  Here, $p_{ij} = \frac{c_i(s_j)}{\sum_j^m c_i(s_j)}$, $j = 1, \ldots, n$, $c_i(s_j)$ is the performance value of alternative s_j and if $p_{ij} = 0$ then $p_{ij} log_2 p_{ij} = 0$.
- Calculate the entropy coefficient weight (w_i) of each criterion i :

$$w_i = \frac{(1 - H_i)}{m - \sum_i^m H_i} \qquad (9)$$

  Here, $0 \le w_i \le 1$, $\sum_{i=1}^{m} w_i = 1$.

Algorithm 1 Election Procedure (EP)

1: **Input:** S: Set of sensor nodes, and k: number of CHs.
2: **Output:** k: sensor nodes with highest product P values.
3: $S(j).c_i$: criterion value of sensor node j for a criteria i
4: m: number of criteria
5: n: number of alternatives or sensor nodes in S
6: **for** $i = 1$ to m **do**
7:     **for** $j = 1$ to n **do**
8:         $p_{ij} = \frac{S(j).c_i}{\sum_{u=1}^{n} S(u).c_i}$
9:     **end for**
10:    $H_i = -\frac{1}{log_2 n} \sum_{j}^{n} p_{ij} log_2 p_{ij}$
11: **end for**
12: **for** $i = 1$ to m **do**
13:    $w_i = \frac{(1 - H_i)}{m - \sum_i^m H_i}$
14: **end for**
15: **for** $j = 1$ to n **do**
16:    $S(j).P = \prod_{i=1}^{m} (S(j).c_i)^{w_i}$
17: **end for**
18: Return top k sensor nodes from S with highest P values.

4.1.3 Cluster head election decision

The election decision of CH is made by executing the following steps (The pseudocode of this procedure is presented in Algorithm 1):

- Using EWC, compute the weight of each criterion.
- Using WPM, compute the product value P of each sensor node (s_j) considering m criteria as:

$$s_j.P = (s_j.c_1)^{w_1} \times (s_j.c_2)^{w_2} \times \cdots \times (s_j.c_i)^{w_i}. \qquad (10)$$

  Here, $w_1, w_2, \ldots, w_i$ are the weights for the criteria c_1, c_2, ..., c_i respectively.
- Sensor nodes with highest P values will be elected as CHs.

4.1.4 Cluster heads election procedure

1. *Initial step* In EBCS, the BS finds the optimal number k of CHs in the first round $(r = 1)$ for the WSN. BS participates in selection of CHs only in the first round. The election process will be performed by executing the selection procedure (Algorithm 1) considering the criteria, E, $N_{support}$, D, and DN, i.e., BS call $EP(S, k)$ procedure where S is the set of sensor nodes in WSN.

   BS sends BE_{CH} message to k sensor nodes that have highest P values. After the sensor node receiving the BE_{CH} message, the selected CHs announce their role using CSMA [12, 16, 38] and then each CH forms its cluster by the cluster formation step below.

2. *CH election step* The consumed energy by the CH involves the consumed energy of receiving data from all the CMs, aggregating data, and forwarding data to the BS. As a result, we employ energy prediction technique to predict the CH failure due to energy depletion. If we assume that n sensor nodes are uniformly dispersed and there sensor exist k clusters in the topology. Thus, on average, there are n / k sensor nodes per cluster (one CH and (n / k)-1 CMs). Consequently, the total energy consumption by the CH (e_{CH}) for single round can be calculated as follows [12]:

$$e_{CH} = \left(\frac{n}{k} - 1\right).E_{Rx}(l) + \frac{n}{k}.l.E_{DA} + E_{Tx}(l, d_{toBS}). \qquad (11)$$

   The CM nodes only need to transmit data to its corresponding CH. Thus, the total energy consumption by the CM node (e_{CM}) during one single frame can be calculated as follows:

$$e_{CM} = E_{Tx}(l, d_{toCH}). \qquad (12)$$

   Here, d_{toBS} is the mean distance between the CH and the BS, and d_{toCH} is the average distance between CMs and the CH. In an area with size $M \times M$, d_{toBS} and d_{toCH} can be estimated as:

$$d_{toBS} = \frac{M}{\sqrt{2\pi k}}, \quad d_{toCH} = 0.765\frac{M}{2}. \tag{13}$$

If each sensor node s takes t_{cm} times to work as CM node and t_{CH} times to work as CH node, then, the total energy consumption for sensor node s at round r can be calculated as follows:

$$E_{consumed}(s, r) = t_{ch}e_{CH} + t_{cm}e_{CM} \tag{14}$$

According to the current energy of sensor node s and Eq. 14, the remaining energy of sensor node s at the start of round $r + 1$ can be used to predict when r round ends as follows:

$$E_{prediction}(s, r + 1) = E_{residual}(s, r) - E_{consumed}(s, r). \tag{15}$$

Sensor node s will decide if its current residual energy is close to the predicted energy value at the end of the round by using the following equation:

Algorithm 2 EBCS (BS Side)

1: r: The current round.
2: S: Set of sensor nodes.
3: k: The optimal number of CHs
4: **if** $r = 1$ **then**
5:     BS executes the election procedure by calling $EP(S, k)$
6:     BS transmits BE_{CH} message to the k sensor nodes that have highest product values Ps.
7: **end if**
8: Waits for data from network

$$\upsilon = |1 - \frac{E_{prediction}(s, r)}{E_{residual}(s, r)}|. \tag{16}$$

The error of energy prediction can be tolerated if υ is less than a constant ϵ.

At the end of each round r and based on the received members' information (id, $E_{prediction}$, $E_{residual}$, $N_{support}$), CH executes Algorithm 1 in order to take a decision for the next round. CH has three cases (Algorithm 3, steps 26-38):

(a)   The list of sensor nodes to be nominated as CHs (ψ) is empty that means none of CMs or current CHs can be CH for the next round because the energy needed to act as CH e_{CH} is larger than the predicted energy $E_{prediction}$. In this case, the current CHs inform their CMs to send their data directly to the BS.

In such a case, we avoid unreliable and unpredicted behavior of the network by avoiding forming clusters. However, the positive aspect is that with the remaining energy of a sensor node, it may succeed if data is sent directly to the BS.

(b)   If ψ contains only one CH, then the CH will remain working as CH for the next round.

(c)   If ψ contains more than one CH ($|\psi| > 1$), as a result CH has to execute the election procedure EP with the set ψ ($EP(\psi, 1)$) to make a decision.

If the cardinality of ψ is larger than one, then the CH sensor node has a set of alternatives. The first alternative is to stay working as CH sensor node, and the second alternative is to select one of its CMs to work as CH.

CH executes the election procedure EP considering ψ members and the four criteria $N_{support}$, E, D, and DN, then the product value of each member node is computed and compared, then the sensor node with the highest product value P will be selected as the next CH (Algorithm 3, Step 34).

4.2 Cluster formation phase

Each CH sends BE_{CH} message to the elected CHs. The elected CHs announce their role using CSMA. The advertising short message $CH_{Announce}$ contains the ID of CH. Then, cluster formation step is executed (Algorithm 4).

To ensure that the concise information will be sent to BS, each non-CH node selects the CH that uses minimum communication cost, based on the signal strength of received advertisement and then transmits $JOIN_{Request}$ message to the selected CH.

The CHs build a TDMA schedule or transmission plan that decides the time for each CM to send its collected data. This guarantees collisions avoidance between data messages and permits radio components of CMs to be turned on only during their transmission time [12, 38]. Creating and transmitting TDMA schedule will be the last step in setup phase.

4.3 Data transmission phase

Normally, each CM sends only its collected data to the corresponding CH in its allocated time slot (Algorithm 4,

Step 8–11). However, in our proposed EBCS, the data message includes the ID of CM, residual energy, $N_{support}$ and $E_{prediction}$ data, i.e., the local information of the CMs can be employed to localize the CH rotation in the subsequent rounds.

Once all data is received by the CH, it executes some signal processing function such as data aggregation. Then, the aggregated result is sent to the BS. At the end of this phase, using piggybacked CM information, each present CH decides whether it would continue as CH or give up its role (Algorithm 3, Steps 17–38).

Algorithm 3 EBCS (CH side)

1: $G_{CM}(ch)$: List of CMs of CH.
2: ψ: List of sensor nodes to be nominated as CHs.
3: r: The current round.
4: S: Set of sensor nodes.
5: **if** BE_{CH} message received **then**
6:     Advertise your role as CH by transmitting $CH_{Announce}$ message
7:     Wait for $JOIN_{Request}$ messages from neighbor sensor nodes
8: **end if**
9: **if** $JOIN_{Request}$ message received from s_j **then**
10:     Add sensor node s_j to $G_{CM}(s_i)$
11: **end if**
12: Transmit TDMA schedule to member sensor nodes.
13: {Data transmission phase.}
14: **if** Data received from $G_{CM}(s_i)$ members **then**
15:     Perform data fusion on data and send the result to the BS
16: **end if**
17: Calculate $E_{prediction}(s_i, r + 1)$ and $N_{support}(s_i)$
18: **if** $e_{CH} < E_{prediction}(s_i, r + 1)$ **then**
19:     Add ch to ψ
20: **end if**
21: **for** each sensor node $cm_i \in G_{CM}(s_i)$ **do**
22:     **if** $e_{CH} < E_{prediction}(cm_i, r + 1)$ **then**
23:         Add cm_i to ψ
24:     **end if**
25: **end for**
26: **if** $|\psi| = 0$ **then**
27:     CH informs $G_{CM}(s_i)$ members to send directly to BS.
28:     Relinquish your role as CH and send directly to BS
29: **else**
30:     **if** $|\psi| = 1$ and $s_i \in \psi$ **then**
31:         s_i decides to remain working as CH.
32:     **else**
33:         **if** $|\psi| > 1$ **then**
34:             s_i executes the election procedure by calling $EP(\psi, 1)$
35:             s_i transmits BE_{CH} message to the next CH node.
36:         **end if**
37:     **end if**
38: **end if**

Algorithm 4 EBCS (None CH sensor node side)

1: **if** $CH_{Announce}$ message received from s_j **then**
2:    send $JOIN_{Request}$ to s_j that requires minimum communication cost,
3:    wait for TDMA schedule message.
4: **end if**
5: **if** TDMA schedule message received **then**
6:    wait for your time slot
7: **end if**
8: **if** time slot is true **then**
9:    Calculate $E_{prediction}$ and $N_{support}$
10:    Transmit aggregated data to the CH along with (ID, $E_{prediction}$, $E_{residual}$, $N_{support}$)
11: **end if**
12: **if** direct to BS message received **then**
13:    Transmit aggregated data to BS directly
14: **end if**

4.4 Entropy based clustering algorithm analysis

Each CH gathers and aggregates the data from its CMs, and then forward the aggregated data to BS. As a result, the consumed energy by CH is more than the consumed energy by CM. In EBCS, the sensor nodes $CH_{Announce}$ with greater product value of the sensor node with greater remaining energy, greater number of neighbors, and the nearest to BS will more plausible to be elected as a CH. Moreover, every non-CM that receives $CH_{Announce}$ adopts the CH that needs minimum communication cost based on the distance between it and the CH candidates. This will balance the network load during the CH rotation in successive rounds.

Lemma 1 *The proposed EBCS has $O(n)$ exchanged messages as an overhead, where n is the number of sensor nodes.*

Proof If k is the average number of CHs per round. In each round we will have the following overhead messages:

- k messages for broadcasting $CH_{Announce}$ by all CHs.
- $n - k$ messages for $JOIN_{Request}$ by non-CMs.
- k messages for broadcasting TDMA schedule by all CHs.
- $n - k$ messages for sending data to CHs by CMs.
- k messages for sending data to BS by CHs.
- Finally, in the worst case, k messages for the rotation of CHs (it will take place where each CH determines to stay as CH or select one of its members to be the successive CH).

Therefore, the EBCS overhead will be $O(n)$ as $k \ll n$. □

5 Simulation results

Our algorithm is simulated using MATLAB R2015a. 100 sensors are randomly deployed in our simulations in the following regions $100 \times 100, 150 \times 150, 200 \times 200, 250 \times 250, 300 \times 300, 350 \times 350\,\mathrm{m}^2$ on a two-dimensional plane with BS placed at the center. The radio model and energy parameters are used as previously described in Sect. 3. The main parameters and their values of the simulation are provided in Table 2. The performance metrics we pursue here are as follows:

1. *Stability period* It is the duration from the start of a network simulation till the time first sensor node dies (FND).
2. *Half-life* Time from the beginning of WSN deployment to a time when a sensor node consumes up to half of its energy.
3. *Consumed energy* Shows how much energy has been consumed by the network with increasing number of rounds.
4. *Number of alive sensor nodes per round* The number of alive sensor nodes in the WSN after each round.

The algorithm has been tested using different random topologies and it is compared with DEC, SEP-E, EDDEEC [32], MEDDEEC[35], CREEP [24] and DMSEP [29] algorithms. The optimal parameters of EBCS with DEC, SEP-E, EDDEEC, MEDDEEC, CREEP, DMSEP are utilized to achieve the best performance. We assume that the total energy of the network is 102.5 J for all algorithms. Table 3 shows the ratio of sensor nodes and their corresponding energies.

In the next sections, we explore the performance metrics and compare the results of the proposed algorithm with many existing baseline algorithms.

5.1 Analysis of optimum cluster heads

In EBCS, the clustering algorithm was created to guarantee that the expected number of CHs per round is fixed, which can be set a prior. we use a probabilistic-based model such as EDDEEC, MEDDEEC, CREEP, DMSEP and SEP-E and non-probabilistic-based such as DEC to analyze the number of elected CHs per round. A number of experiments have been conducted by varying the number of CHs

Table 2 Simulated parameters

Parameter	Value
Network area size	100×100, 150×150, 200×200, 250×250, 300×300, and 350×350
Nodes	100
Initial energy	0.5 J, 1.25 J, 2.0 J
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{mp}	0.00013 pJ/bit/m^4
d_0	87 m
E_{DA}	5 nJ/bit/signal
Packet size	4000 bits
Percentage of CHs Pop.	0.1

Table 3 Energy setting of sensor nodes

% of nodes	Energy (J)
20% of the nodes	2
30% of the nodes	1.25
50% of the nodes	0.5

per round for each Algorithm. Figure 4 shows the number of CHs for the first 2500 rounds in EBCS, DEC, EDDEEC, MEDDEEC, SEP-E, CREEP, and DMSEP algorithms. We notice that the number of CHs per round for EDDEEC, MEDDEEC, CREEP, DMSEP and SEP-E fluctuate which indicates that the optimal number of elected CHs cannot be

guaranteed in each round as they depend on probability function for CH selection). Due to this, in some rounds very few CHs are elected, and the cluster-members will need to transmit at a much longer distances to reach their CHs. Likewise, if the number of elected CHs is higher, not much data aggregation will be done, since the cluster size is smaller. Thus, more energy will be used for transmitting. This is one of the major drawback of the probabilistic-based model. Another disadvantage of this model is that, the energy consumption across nodes become increasingly uneven as the network progresses. While this is not the case for DEC and EBCS. This ensures that for any network size, once the optimal number of CHs is defined in the beginning of the network operation, the number of elected CHs will remain fixed that evenly distributes energy dissemination as the network evolves.

5.2 Stability period

The stability period is defined as the duration from the start of the network simulation till the time first sensor node dies. Figures 5 and 6 show the following points:

1. The stability period of the proposed algorithm exceeds other algorithms as the remaining energy is predicated and cluster load is considered during CH selection process. Moreover, using entropy enhances the stability period of the network by selecting the nodes with more weight according to their energy level, etc.
2. The stability period of all algorithms degrades as the network size increases, where the distance between
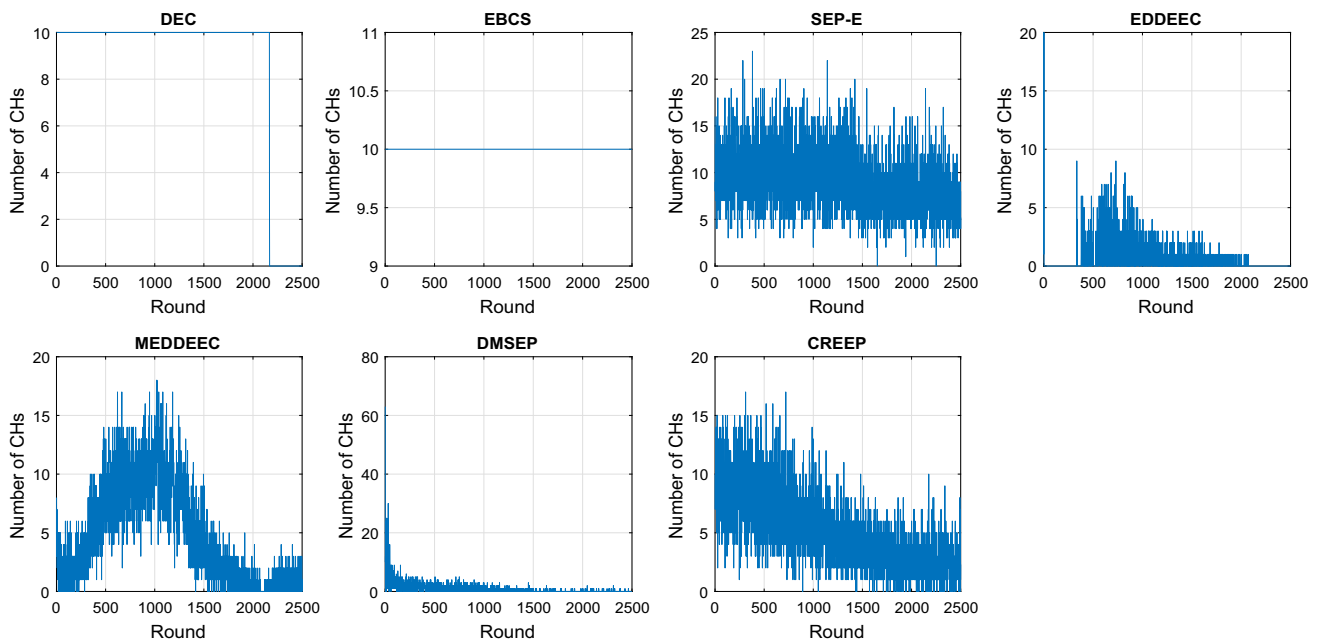
Fig. 4 Number of clusters per round in DEC, SEP-E, EDDEEC, EBCS, MEDDEEC, DMSEP, and CREEP algorithms
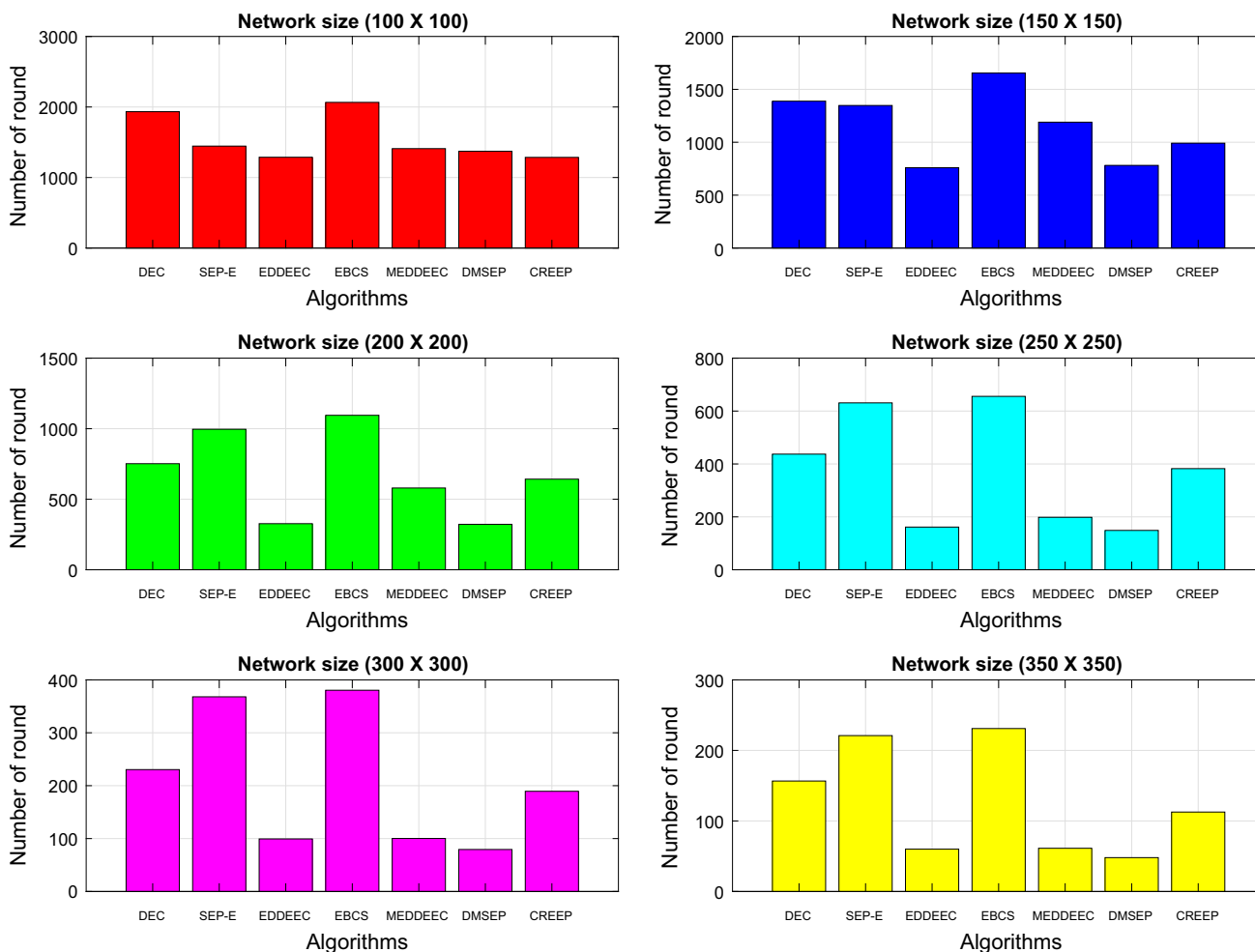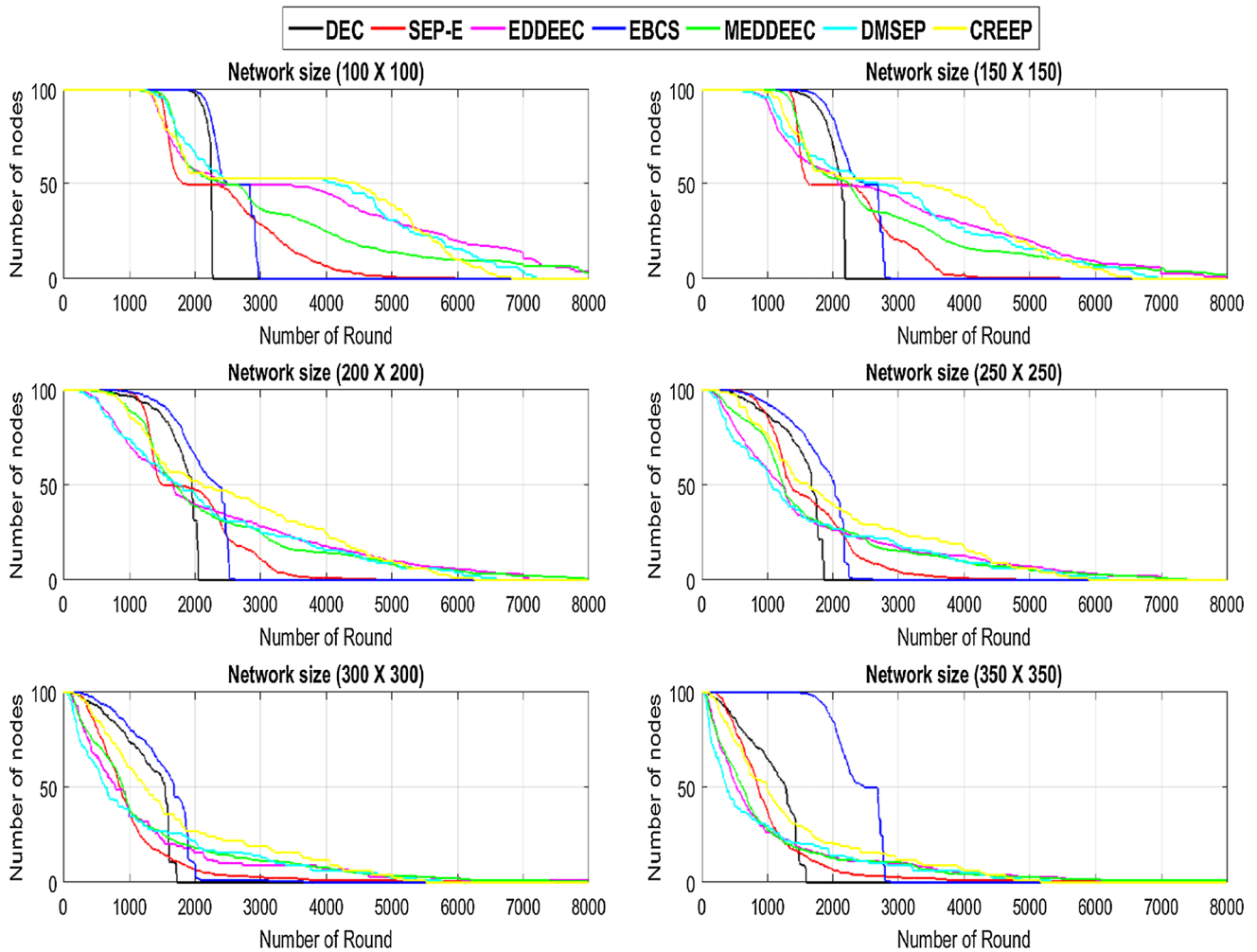
Fig. 5 Stability period for DEC, SEP-E, EDDEEC, EBCS, MEDDEEC, DMSEP, and CREEP algorithms for different network sizes

sensor nodes increases and so the consumed energy for sending and aggregating data increases.

3. The stability period of DEC exceeds the stability period of SEP-E by 35% for the network size of 100×100. But, as the network size increases from 200×200 to 350×350, stability period of SEP-E exceeds the stability period of DEC by 25%, 31%, 37%, 29% respectively, because the DEC depends on the sensor node energy as selection parameter for CHs as network size expands the distance between sensor nodes increases and so the consumed energy increases while the selection processes in SEP-E is probabilistic.

4. The stability period of MEDDEEC, DMSEP and EDDEEC significantly degrades as the network size increases to 350×350 because MEDDEEC and EDDEEC depend on the absolute threshold value (T absolute) that is calculated in each round, and the energy of all sensor nodes are compared with the threshold value. Only if the energy of a sensor node is less than the threshold, it will be treated independently.

The value of the absolute threshold does not depend on the network size which affects the network lifetime as the network size increases. While, DMSEP considers distance threshold value that is calculated as the average distance between nodes and BS. However, the threshold value is calculated at the network start up and fixed in all rounds which affects the network lifetime too. As the network size increases to 350×350, the proposed algorithm enhances the stability period compared with DEC, EDDEEC and MEDDEEC up to of 32%, 74%, 73%, respectively.

5. EDDEEC calculates the probability for CH selection based on the initial energy and remaining energy level of sensor nodes and average energy of the network while MEDDEEC considers average distance and the distance from BS as parameters to compute the probability function that allows MEDDEEC to select a sensor node to be CH only if it is closer to the base station and hence minimizing the energy consumption is reduced. As a result the performance of MEDDEEC

Fig. 6 Number of alive sensor nodes per round of DEC, SEP-E, EDDEEC, EBCS, MEDDEEC, DMSEP, and CREEP algorithms for different network sizes

is better than EDDEEC in terms of energy consumption.

6. As the network size increases to 250×250 and 300×300, the stability period of SEP-E increases slightly by 1% over EBCS for network size. However, as the network size increases to 350×350 the stability period of EBCS is increased by 5% over SEP-E.

In SEP-E, there are three reasons that lead to a reduction in network lifetime

- In SEP-E, CHs selection depends on the probability of each type of sensor node without considering the remaining energy of each sensor node so that the sensor nodes with comparatively small remaining energy can be CHs.
- As we discussed in previous Sect. 5.1, SEP-E does not have stable number of CHs, i.e., the optimal number of CHs is not guaranteed in many rounds.

- In case of increasing network size, at the edge of the network or in low density regions, the number of CHs increases and so many cluster members inefficiently utilize energy while communicating with the CHs.

For the above reasons, stability period of SEP-E becomes very close to EBCS as the network size increases to specific size (300×300), then after this specific size, the EBCS will start to exceed SEP-E.

5.3 Half-life time

Our algorithm enhances the Half-life of WSN better than DEC, SEP-E, EDDEEC, MEDDEEC, CREEP and DMSEP for different network sizes as the weights are updated in each round for the criteria and are calculated based on the current status of sensor nodes. Moreover, we try not to overload the sensor node that becomes CH by considering cluster load as the selection criteria and assuming if it can continue its task until the end of the round. This leads to
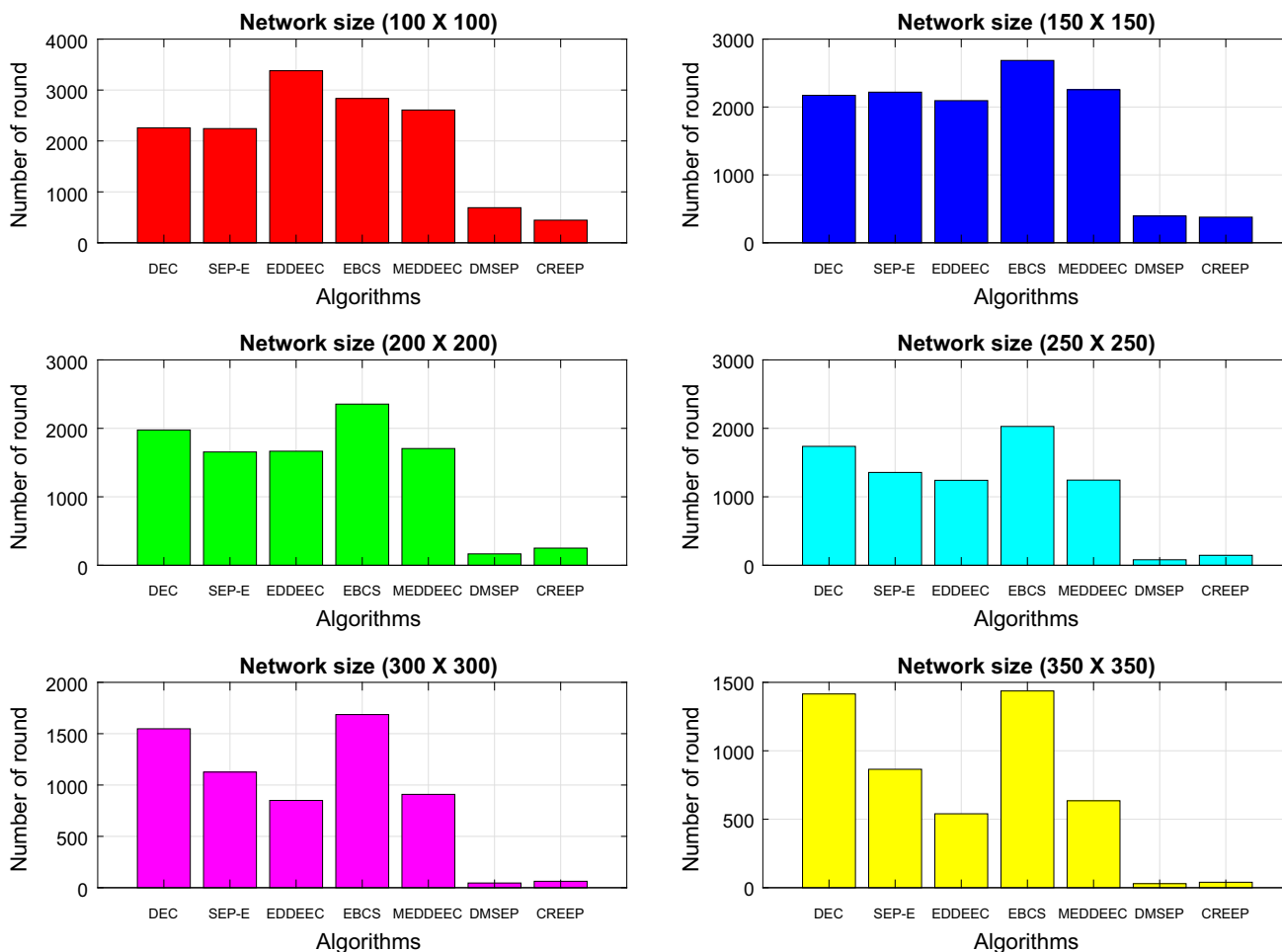
Fig. 7 Half-life ofDEC, SEP-E, EDDEEC, EBCS, MEDDEEC, DMSEP, and CREEP algorithms for different network sizes

energy saving for sensor nodes that have low energy levels as they work as normal sensor nodes and do not handle the CH tasks. Figure 7 shows that the proposed algorithm enhances the WSN Half-life compared with DEC, SEP-E, EDDEEC, MEDDEEC, CREEP, and DMSEP up to a magnitude of 47%, 45%, 48% , 88%, 128% and 47% respectively for network size of 100×100.

The results show that the network lifetime of SEP-E increases slightly by 1% over EBCS for network size 250×250 and 300×300. However, as the network size increases to 350×350, the stability period of EBCS increases by 5% over SEP-E. This is because the number of CHs in each round is not fixed due to its randomized behavior in SEP-E; so the chance of selecting fewer

number of CHs than the required and near to the BS increases. This leads to some indirect energy saving. Overall, our proposed EBCS increases the WSN lifetime as compared with DEC, SEP-E, EDDEEC, MEDDEEC, CREEP, DMSEP up to a magnitude of 7%, 43%, 60%, 47%, 61%, 50%, respectively for network size of 100×100.

5.4 Average remaining energy

We compute the average standard deviation of the residual energy of sensor nodes per round with different network sizes. Figure 8 shows the average of remaining energy per round for DEC, SEP-E, EDDEEC, EBCS, MEDDEEC,
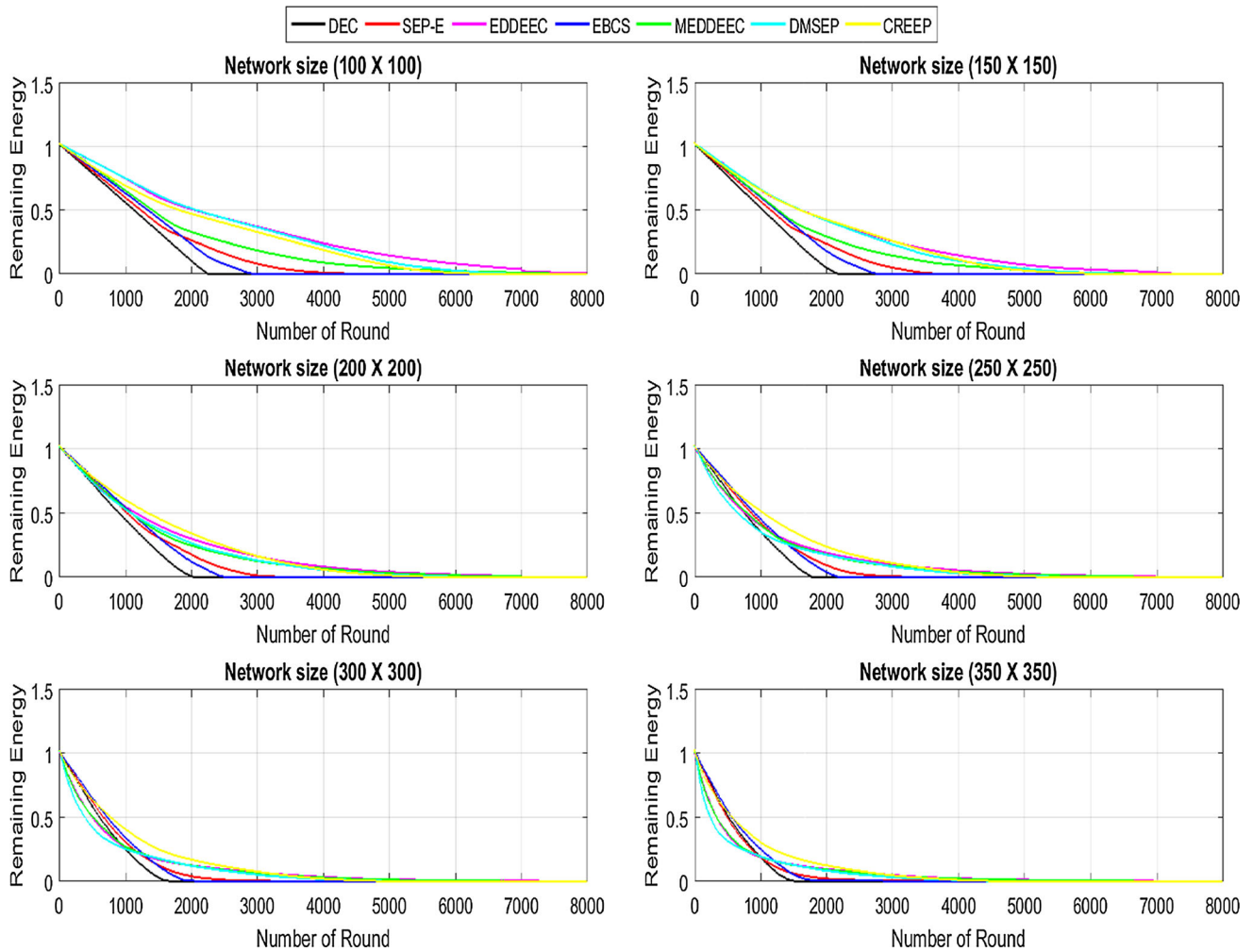
Fig. 8 Average residual energy per round versus network size for DEC, SEP-E, EDDEEC, EBCS, MEDDEEC, DMSEP, and CREEP algorithms
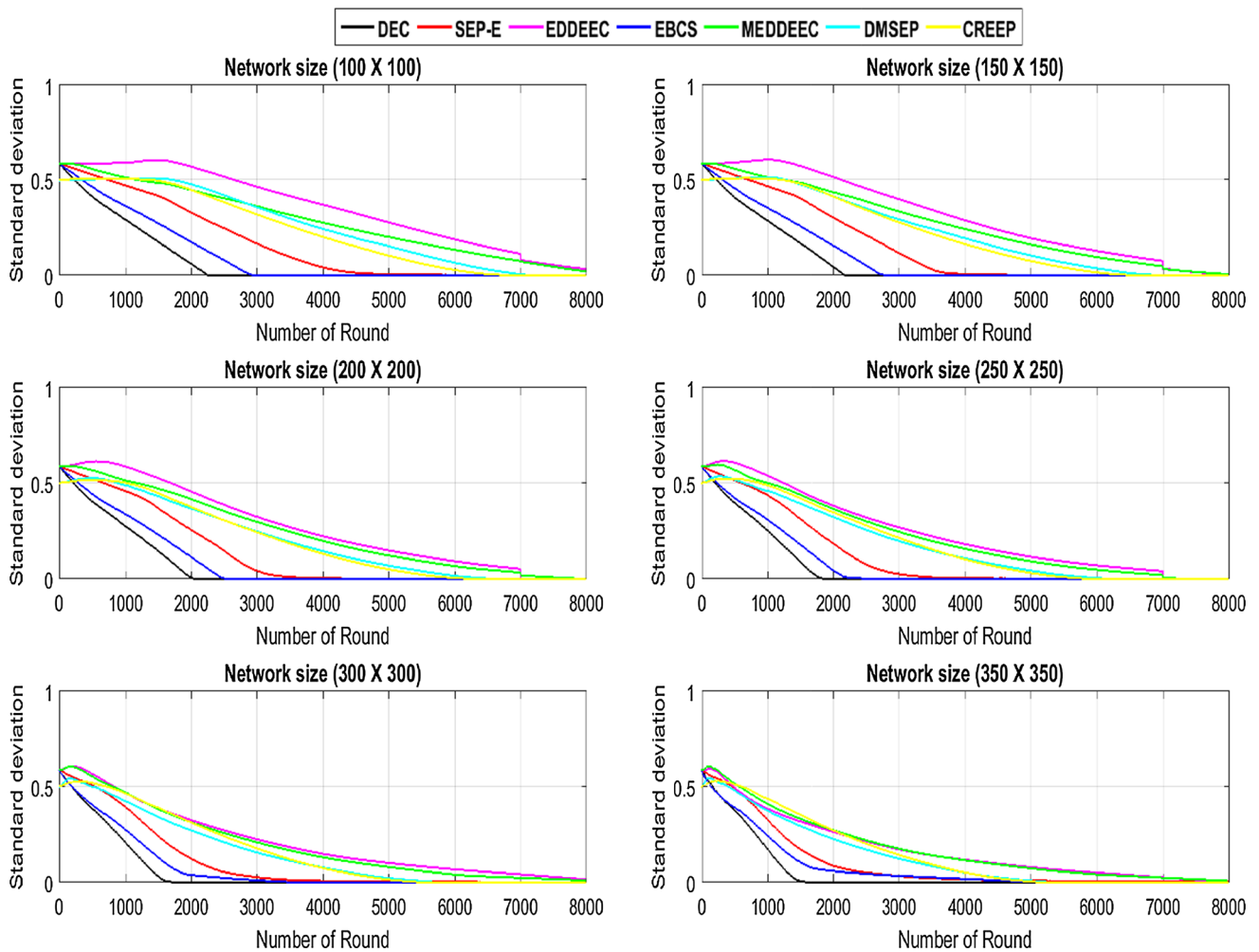
CREEP, DMSEP algorithms. Notice that as the network size expands the consumed energy for SEP-E and EBCS slightly raises, while there is a remarkable increase in the case of DEC, EDDEEC, and MEDDEEC.

Figures 8 and 9 show the average and standard deviation of residual energy of sensor nodes as the number of rounds increase. The standard deviation of residual energy of sensor nodes refers to the distribution rate of residual energy of sensor nodes. A small standard deviation represents that the residual energy of each sensor node is almost uniform and is similar to the average value. The standard deviations for the residual energy among the sensor nodes of each protocol is computed as shown in

Fig. 9. The rapid decrease in both of DEC and EBCS curves indicates that DEC and EBCS balance the energy consumption in the network better than the other protocols. This is due to even distribution of consumed energy among the sensor nodes.

6 Conclusion

In this paper, we have proposed a new entropy based clustering algorithm (EBCS) which divides a WSN into two-levels of the hierarchical network and deals with three-levels of energy heterogeneity of sensor nodes. The

10. Triantaphyllou, E. (2000). *Multi-criteria decision making methods*. New York: Springer.

11. Bhunia, S. S., Das, B., & Mukherjee, N. (2014). EMCR: Routing in WSN using multi criteria decision analysis and entropy weights. In *Internet and distributed computing systems, IDCS 2014, lecture notes in computer science* (Vol. 8729). Cham: Springer.

12. Rabiner Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii international conference on system sciences* (pp. 1–10).

13. Rabiner Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1, 660–670.

14. Khediri, S. E., Nasri, N., Wei, A., & Kachouri, A. (2014). A new approach for clustering in wireless sensors networks based on LEACH. *Procedia Computer Science*, 32, 1180–1185.

15. Handy, M. J., Haase, M., & Timmermann, D. (2002). Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In *4th international workshop on mobile and wireless communications network* (pp. 368–372).

16. Aderohunmu, F. A., Deng, J. D., & Purvis, M. K. (2011). A deterministic energy efficient clustering protocol for wireless sensor networks. In *Proceedings of the seventh IEEE international conference on intelligent sensors, sensor networks and information processing (IEEE-ISSNIP)*, Adelaide, Australia (pp. 341–346).

17. Smaragdakis, G., Matta, I., & Bestavros, A. (2004). SEP: A stable election protocol for clustered heterogeneous wireless sensor networks. In *Proceeding of the international workshop on SANPA*.

18. Kumar, D., Aseri, T. C., & Patel, R. B. (2009). EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks. *Computer Communications*, 32(4), 662–667.

19. Salim, A., Osamy, W., & Khedr, A. M. (2014). IBLEACH: Effective LEACH protocol for wireless sensor networks. *Wireless Networks*, 20, 1515–1525.

20. Sharma, S., Bansal, R. K., & Bansal, S. (2017). Heterogeneity-aware energy-efficient clustering (HEC) technique for WSNs. *KSII Transactions on Internet and Information Systems*, 11(4), 1866–1888.

21. Fu, C., Jiang, Z., Wei, W. E. I., & Wei, A. (2013). An energy balanced algorithm of leach protocol in WSN. *International Journal of Computer Science*, 10(1), 354–359.

22. Amodu, O. A., Azlina, R., & Mahmood, R. (2018). Impact of the energy-based and location-based LEACH secondary cluster aggregation on WSN lifetime. *Wireless Networks*, 24, 1379–1402.

23. Mostafa Bozorgi, S., & Massoud Bidgoli, A. (2018). HEEC: A hybrid unequal energy efficient clustering for wireless sensor networks. *Wireless Networks*. https://doi.org/10.1007/s11276-018-1744-x.

24. Dutt, S., Agrawal, S., & Vig, R. (2018). Cluster-head restricted energy efficient protocol (CREEP) for routing in heterogeneous wireless sensor networks. *Wireless Personal Communications*, 100, 1477–1497. https://doi.org/10.1007/s11277-018-5649-x.

25. Dutt, S., Kaur, G., & Agrawal, S. (2019). Energy efficient sector-based clustering protocol for heterogeneous WSN. *Proceedings of 2nd international conference on communication, computing and networking, lecture notes in networks and systems*

26. Sharma, D., Ojha, A., & Bhondekar, A. P. (2018). Heterogeneity consideration in wireless sensor networks routing algorithms: A review. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227-018-2635-8.

27. Wang, Z.-X., Zhang, M., Gao, X., Wang, W., & Li, X. (2017). A clustering WSN routing protocol based on node energy and multipath. *Cluster Computing*. https://doi.org/10.1007/s10586-017-1550-8.

28. Singh, D., & Panda, C. K. (2015). Performance analysis of modified stable election protocol in heterogeneous WSN. In *International conference on electrical, electronics, signals, communication and optimization* (p. 15).

29. Singh, A., Singh Saini, H., & Kumar, N. (2019). D-MSEP: Distance incorporated modified stable election protocol in heterogeneous wireless sensor network. In *Proceedings of 2nd international conference on communication, computing and networking, lecture notes in networks and systems* (p. 46).

30. Qing, L., Zhu, Q., & Wang, M. (2006). Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer Communications*, 29(12), 2230–2237.

31. Saini, P., & Sharma, A. K. (2010). E-DEEC-enhanced distributed energy efficient clustering scheme for heterogeneous WSN. In *First international conference on parallel, distributed and grid computing (PDGC 2010), Solan* (pp. 205–210).

32. Javaid, N., Rasheed, M. B., Imran, M., Guizani, M., Khan, Z. A., Alghamdi, T. A., et al. (2015). An energy-efficient distributed clustering algorithm for heterogeneous WSNs. *EURASIP Journal on Wireless communications and Networking*, 2015, 151.

33. Singh, S., Malik, A., & Kumar, R. (2017). Energy efficient heterogeneous DEEC protocol for enhancing lifetime in WSNs. *Engineering Science and Technology: An International Journal*, 20(1), 345–353. https://doi.org/10.1016/j.jestch.2016.08.009.

34. Javaid, N., Qureshi, T. N., Khan, A. H., Iqbal, A., Akhtar, E., & Ishfaq, M. (2013). EDDEEC: Enhanced developed distributed energy-efficient clustering for heterogeneous wireless sensor networks. *Procedia Computer Science*, 19, 914–919.

35. Shaji, M., & Ajith, S. (2015). Distributed energy efficient heterogeneous clustering in wireless sensor network. *2015 fifth international conference on advances in computing and communications (ICACC), Kochi* (pp. 130–134).

36. Mazumdar, N., & Om, H. (2017). DUCR: Distributed unequal cluster based routing algorithm for heterogeneous wireless sensor networks. *International Journal of Communication Systems*, 30, e3374. https://doi.org/10.1002/dac.3374.

37. Han, R., Yang, W., Wang, Y., & You, K. (2018). DCE: A distributed energy-efficient clustering protocol for wireless sensor network based on double-phase cluster-head election. *Sensors*, 17(5), 998.

38. Aderohunmu, F. A., Deng, J. D., & Purvis, M. K. (2011). Enhancing clustering in wireless sensor networks with energy heterogeneity. *International Journal of Business Data Communications and Networking*, 7(4), 18–32.

Walid Osamy received his B.Sc. degree with honor in Computer Science in June 2000, the M.S. degree in June 2006, and the Ph.D. degree in September 2010 all in area of computer science and from the Faculty of science, Zagazig University, Egypt. He has been involved in the projects of network infrastructure, and management information systems at Communication Information Technology Center (CITC), Zagazig University, Egypt. From December 2010 till now he has been working as Assistant Professor at the Department of Computer Science, College of Computers and Informatics, Benha University (http://www.fci.bu.edu.eg/), Egypt. From March 2015 until now, he has been working as Assistant Professor at Qassim University, Buridah, KSA. His research interests include computational intelligence and in the field of IoT [mobile computing, wireless sensor networks (WSNs)].

Ahmed Salim received his B.Sc. degree in Computer Science in June 2001, and the M.Sc. degree in the area of distributed computing in December 2006, both from Zagazig University, Egypt. In October 2010 he received his Ph.D. degree in systems, network and telecommunication devices from Bonch-Bruvich University of Telecommunication, Saint-Petersburg, Russia. From January 2012 to December 2012, he was a Consultant of Information and Communication Technology Project (ICTP) at Zagazig University, Egypt. From January 2011 to January 2014, he was an Assistant Professor at the Department of Mathematics, Faculty of Science, Zagazig University, Egypt. From January 2014 till now, he has been working as Assistant Professor at the Department of Mathematics, Faculty of Science and Art, Qassim University, Al-mithnab, Kingdom of Saudi Arabia. His research interests include decomposable algorithms, computing, wireless networks.

Ahmed M. Khedr received his B.Sc. degree in Mathematics in June 1989 and the M.Sc. degree in the area of optimal control in July 1995 both from Zagazig University, Egypt. In July 1999 he received his M.Sc. and in March 2003 he received his Ph.D. degree both in area of Computer Science and Engineering from University of Cincinnati, Ohio, USA. From March 2003 to January 2004, he was a Research Assistant Professor at ECECS Department University of Cincinnati, USA. From January 2004 to May 2009, he worked as Assistant Professor at Zagazig University, Egypt, from September 2009 to September 2010 he worked as Associate Professor at the Department of Computer Science, College of Computers and Information Systems, Taif University, KSA, from September 2010 till now he is working as Associate Professor at the University of Sharjah, UAE, and from September 2014 till now he is working as a Professor at the Department of Mathematics, Faculty of Science, Zagazig University, Egypt. In June 2009, he awarded the State Prize of distinction in advanced technology. In May 2013, he awarded the Sharjah Islamic Bank prize of distinction in research. In April 2015, he awarded the University of Sharjah Prize of distinction in research. He has coauthored 65 works in journals and conferences relating with optimal control, wireless sensor networks, internet of things, distributing computing, and bioinformatics.