IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 September 2022, accepted 3 October 2022, date of publication 10 October 2022, date of current version 17 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213037

RESEARCH ARTICLE

Multi-Criterion Partial Clustering Algorithm for Wireless Sensor Networks

PEYMAN NEAMATOLLAHI
Computer Engineering Department, Faculty of Computer and Electrical Engineering, Hakim Sabzevari University, Sabzevar 9617976487, Iran
e-mail: p.neamatollahi@hsu.ac.ir

ABSTRACT Cluster architectures are an effective approach for organizing sensor networks to balance the load and prolong network life. To cluster wireless sensor networks, this paper proposes an energy-efficient distributed algorithm. This algorithm uses two techniques (partial clustering and multi-criterion cluster formation) for efficient use of the sensor nodes' energy. When a header expends a certain amount of power, it only notifies the nodes in its cluster that new clustering is required in the next round. Therefore, in contrast to previous studies that performed complete clustering, clustering in the present work is performed partially, which considerably reduces the clustering overhead. In addition, a multi-criterion score is calculated for each node. In this algorithm, a node with the highest remaining energy and score is a more suitable candidate to be selected as the head of the cluster. In addition, a regular node becomes the member of the cluster with the highest score in its vicinity. The experiments reveal the superiority of the proposed algorithm over other simulated algorithms in terms of energy savings and network lifetime.

INDEX TERMS Sensor networks, clustering, network lifetime, energy efficiency, distributed algorithm.

I. INTRODUCTION

Wireless sensor networks (WSNs) provide reliable remote monitoring. These networks are commonly used as data collection networks. In these networks, the sensed data are highly correlated, and the end-user requires high-level information from the raw data [1], [2]. The main task of any sensor node in these networks is to detect events, quickly process local data, and transmit information [3]. In applications where nodes are left in the sensor environment, they have low mobility and are limited in their energy and processing power [3], [4], [5], [6]. Recharging the battery of sensor nodes may be impossible or costly, therefore, all design aspects related to sensor networks, from hardware to protocols, should be highly energy efficient [2], [7].

In cluster-based WSNs, network nodes are split into clusters. The task of each sensor node is to transmit the information received from the environment to the head of its cluster and the cluster header (CH) transmits the information collected from the members of its cluster to the Base Station (BS). The headers significantly decrease the amount of data transmitted to the BS by aggregating and merging the raw data received from their member nodes, resulting in savings in bandwidth and network energy resources [8], [9], [10]. When clusters are created, each sensor node is given a specific time slot. Therefore, every member node knows when to send its data. As a result, except for its specific time interval, a node can sleep during the remaining Time Division Multiple Access (TDMA) frame. In other words, it just needs to be active during its specific time interval [11], [12]. In short, using a common timeline, clustering coordinates nodes to transfer their data in the steady-state phase, thus eliminating overhearing, collisions, and idle listening. As a result, clustering results in a significant reduction in the energy dissipation of the nodes [8], [13], [14]. In addition, clustering allows the network to be scalable into hundreds or thousands of nodes. In many applications, clustering is used as a common solution to take advantage of correlation and eliminate the redundancy of data received from sensors. However, despite these advantages, clustering leads to significant overhead due to the exchange of clustering messages.

In this paper, a Multi-criterion Partial Clustering Algorithm (MPCA) for WSNs is presented. In this algorithm, based on local information, each node decides whether to become

The associate editor coordinating the review of this manuscript and approving it for publication was Pallab K. Choudhury.

a CH or join a cluster. The nodes have a partial view of the sensor network. In other words, the algorithm is distributed. Clusters created based on this algorithm are not fixed and change over the lifetime of the network, i.e. clustering is done dynamically. In this algorithm, nodes use an iterative process to decide their status. Whenever a node either recognizes a header to join or becomes a header, it will stop performing its iterative process. The goal of MPCA, as the main contribution of this paper, is to achieve greater energy savings for nodes and thus increase the lifetime of the network. The main properties of MPCA are as follows:

- In contrast to previous approaches that performed complete clustering, this paper recommends partial clustering. In MPCA, partial clustering is performed whenever a CH has consumed a certain portion of its energy. Using a special message, CH notifies member nodes of its inability to continue current responsibilities. Each member that receives this particular message from its cluster header prepares itself for clustering at the beginning of the next round. Using partial clustering, MPCA reduces both the additional overhead caused by the consecutive clusterings and the clustering messages when compared to the previous algorithms.
- In MPCA, the headers are picked based on their residual energy and their score. Hence, each node calculates a multi-criterion score. The amount of the score depends on the number of its neighbors and the centrality of the sensor node between its neighbors. In addition, each regular node chooses a cluster header with the highest score to join.
- The simulation results reveal the superiority of MPCA over the other simulated algorithms in terms of energy savings and lifetime.

The outline of the paper is as follows: Section 2 describes related works. In Section 3, a decentralized energy-aware clustering algorithm is introduced. In Section 4, the simulation results from the comparison of some well-known algorithms and the MPCA algorithm are presented in terms of the number of CH elections, the number of clusters created, the energy consumption, and the network lifetime. In the last section, a conclusion is presented.

II. RELATED WORKS

Previous clustering algorithms in the literature are either static or dynamic. In static clustering [15], [16], clusters are formed once and for all time. Conversely, in dynamic clustering [17], [18], the lifetime of the network is broken down into periods called rounds. Clusters are formed at the beginning of each round and are fixed throughout the round. At the beginning of the next round, clustering is done again. Consecutive reclustering imposes additional overhead on the network nodes.

To cluster WSNs, some current algorithms use various tools such as fuzzy logic or metaheuristic algorithms. For example, TTDFP [19] is a two-tier fuzzy algorithm for node clustering and multi-hop routing in WSNs. TTDFP uses the simulated annealing algorithm to adjust the effective fuzzy parameters in the clustering process. As another example of fuzzy usage in clustering WSNs, the authors in [20] presented a solution that improves the efficiency of fuzzy clustering algorithms by a metaheuristic algorithm.

The clustering algorithms presented in previous research are either centralized or distributed. In the centralized approach [2], [10], [12], [21], it is tried to present a good clustering algorithm relying on different methods and tools such as metaheuristic algorithms. However, centralized algorithms are not efficient in large-scale networks. Because all the necessary information for running a centralized algorithm must be collected in a control center (such as BS), the time and energy of nodes are wasted especially in large networks. In the distributed approach [9], [13], [22], [23], [24], [25], each node independently decides based on local information to join a cluster or become the head of the cluster. Accordingly, such algorithms are more efficient for large-scale networks than centralized algorithms.

From another point of view, distributed algorithms are mostly probabilistic or iterative. Iterative algorithms [18] involve nodes to perform clustering in an iterative process. On the other hand, probabilistic algorithms [17], employ probabilistic methods to select CHs and form clusters. In the following, a brief overview of three leading dynamic distributed clustering algorithms applied in the simulation of this paper is presented.

LEACH [17], [26] is a probabilistic dynamic distributed clustering algorithm. Network operations in LEACH include the setup and steady-state phases. In the setup phase, a random number is selected by each node. The node becomes CH if the produced number is lower than the threshold T(n). The method of calculating T(n) is as follows:

$$T(n) = \begin{cases} \dfrac{p}{1 - p \times \left(r \bmod \frac{1}{p}\right)} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In the above relation, r contains the current round number, p, the desired percentage of the number of CHs, and G includes nodes that have not been converted to CH in the last $1/p$ rounds. If a node is selected as a cluster header, it is necessary to send a message to all network nodes to announce this issue. Upon receiving this notification, each non-CH node selects the nearest CH to join. Each regular node then notifies its CH that it wants to join. Based on the TDMA approach, each CH allocates a time slot to each of its member nodes so that the member node can send the sensed data to its CH in the next stage. During the steady-state phase, each CH, after receiving the data sent from the nodes of its cluster member, aggregates them and sends the aggregated data to BS. In this phase, the main operation of the network is to deliver the received data of the field to the BS. After the end of the steady-state phase, the current round is over and it is necessary to recluster the network at the beginning of the next round. Compared to previous research, the advantages of the LEACH algorithm are:

- Reclustering at the beginning of each round balances the energy consumption of the nodes, thereby increasing the lifetime of the network.
- In the LEACH algorithm, nodes do not need global network information and the algorithm is distributed.

The weaknesses of the LEACH algorithm are:

- When selecting CHs, LEACH does not take into account the residual energy of the sensor nodes.
- Since the selection of CHs is random, it is possible that some CHs are located close to each other and are not well distributed in the field.
- In LEACH, CHs need to transmit their data directly to BS. This is not always possible because BS may not be directly accessible due to factors such as barriers. Therefore, the use of LEACH in large-scale networks is not recommended.

HEED [18], [27] differs from LEACH in how CHs are selected. Both CH selection and cluster formation are based on a combination of the two parameters. The initial parameter depends on the remaining energy of the node. The cost of intra-cluster communications is considered an alternative parameter. In the HEED algorithm, the communication cost is either proportional to 1/node degree, if the production of dense clusters is desirable, or the degree of the node, if the load distribution between CHs is desirable, or AMRP which is defined as the average of the minimum power levels for M nodes in the cluster range that can be accessed CH u, i.e.

$$AMRP\,(u) = \frac{\sum_{i=1}^{M} MinPwr(i)}{M}.$$

In the HEED algorithm, regular nodes select a cluster header to join that has the least communication cost. On the other hand, cluster data is sent to BS by CHs in the form of multi-hop. The advantages of the HEED algorithm are:

- Usually, nodes that are within the range of each other do not become CH concurrently. Therefore, in this algorithm, CHs are well distributed in the environment.
- Multi-hop communication between CHs and BS increases energy savings and scalability.

Transmission distance is a very important factor in node energy waste. To reduce the transmission distance, EHEED [28] (Extended HEED) allows multi-hop communication within the cluster. In other words, regular nodes can send their data to the cluster header through other cluster member nodes. This improvement makes intra-cluster communication cost-effective. However, in this algorithm (similar to LEACH and HEED), performing periodic clustering at the beginning of every round imposes a great energy overhead. The overhead wastes network resources and reduce network lifetime.

III. DESCRIPTION OF THE MPCA ALGORITHM

This section shows the MPCA algorithm and its pseudocode. The MPCA operation consists of rounds, and each round includes the setup phase and the steady-state phase. In the setup stage, CH selection and cluster formation are done. Besides, at this stage, cluster members are scheduled by CH to send their data during the steady-state phase. Regardless of the network diameter, the clustering process ends in a fixed number of iterations. The time interval required for the clustering process is denoted by T_{CP}. At the end of T_{CP}, each sensor node is either an ordinary node that belongs to exactly one cluster or is selected as CH. The steady-state phase consists of TDMA frames. During each frame, each regular node sends the sensed data to its CH at the assigned time slot. At the end of every TDMA frame, the aggregated data is transmitted to BS by each CH via a multi-hop fashion over other CHs. In other words, cluster headers are responsible for intra-cluster coordination and inter-cluster communication. For inter-cluster communication, multi-hop routing is used. To create inter-cluster paths for multi-hop communication with BS, CHs can use an energy-aware routing protocol. In other words, the backbone of the network is formed by CHs, so that packets on CHs are routed in multiple hops from every CH to the BS.

Algorithm 1 Pseudo Code of Local Clustering

1. E_{CH}: Residual energy of the CH node in the end of recent setup phase.
2. α: A floating-point number which is fixed and $0 < \alpha < 1$.
3. **WHILE** steady-state phase is not expired **DO**
4.   **IF** *node is CH* **AND** $E_{residual} < \alpha E_{CH}$ **THEN**
5.   **SEND** *elec − msg* **TO** *its member nodes*
6.   **ELSE-IF** *node is member* **AND** *elec − msg received* **THEN**
7.   The node participates in reclustering for the upcoming round.

A. PARTIAL CLUSTERING

MPCA's innovation is that it does the setup phase partially instead of completely. Also, instead of clustering at the beginning of each round, it does so occasionally, as shown in Algorithm 1. To implement, at the end of each setup step when the clustering process is complete, each CH stores its residual energy in its E_{CH} variable. A partial clustering is required whenever the residual energy of a CH, denoted by $E_{residual}$, during the steady-state phase becomes less than αE_{CH} (α is a constant and $0 < \alpha < 1$). To perform partial clustering, at the end of the current TDMA, a depleted CH sends a message called *elec-msg* to its member nodes. Upon receiving this message, member nodes are prepared for clustering at the beginning of the next round. Therefore, cluster formation and CH election are performed partially rather than completely. This means that all the nodes in the network are not required to participate in the CH election process. Therefore, during T_{CP} interval, some nodes may perform CH election while other nodes wait until this duration finishes. Consequently, the overhead due to frequent complete reclustering is considerably reduced, because clustering is done not only sporadically but also partially. As a result, the energy dissipation of nodes is decreased and the network lifetime is increased.

B. MULTI-CRITERION SCORE

In the MPCA algorithm, the higher the score, the better the chance of being selected as CH. On the other hand, each regular node chooses the CH with the highest score in its vicinity to join. Each node calculates its score based on Equation (2):

$$\text{score}_i = \sum_{j=1}^{\text{num of neighbors}} \frac{1}{\text{dist}^2(\text{i, j})} \tag{2}$$

The two criteria of node degree (the number of neighbor nodes) and node centrality (the centrality of a node among its neighbor nodes) are combined in equation (2) to yield less energy consumption and a longer network lifetime. The effect of these two characteristics on the score are as follows:

The shorter the neighbors' distance from node v, the higher the score_i value. This means that node i is more appropriate for selection as CH if it is more central among its neighbors. The feature of node centrality in CH selection has the following advantages:

- With less power consumption, each regular node can communicate with its CH. Since the energy required to transmit a message is often proportional to the square of the distance, the lower the centrality of the node, the less energy required to send data from other nodes to that node.
- Radio wave interference among CHs is reduced. If the CHs are not in the center of their clusters, some CHs may be in the transmission range of each other and this wave interference leads to the waste of network resources.

The higher the number of neighbors in node v, the higher the score_i value. In other words, under these conditions, node i is more suitable for becoming CH, and therefore, dense clusters are formed. To group spatially close sensor nodes, network splitting into dense clusters is an important goal in many applications. In these applications, the goal is to use data correlation to remove redundant data from sensor readings.

C. CLUSTERING PROCESS

Neighborhood information is updated at the beginning of the MPCA clustering process. Then, each node calculates its score independently. The calculated scores are not distributed to neighbors because they can be exchanged via *CH_msgs* in the following. Note that calculating the score of each node and updating the neighbor information does not have to be done every time clustering triggers. Initially, the optimal ratio of CHs from the total nodes is determined by C_{prob}. Here, C_{prob} is set to 0.05. Each node calculates its probability of becoming a CH, CH_{prob}, (the same as [27]):

$$CH_{prob} = MAX(C_{prob} * \left(E_{residual}/E_{max}\right), p_{min}).$$

In the above relation, E_{max} is the residual energy related to when the battery is fully charged and $E_{residual}$ is the current energy of the node. CH_{prob} should not fall below a certain threshold, p_{min}. The threshold is proportional to the inverse E_{max}. Using the p_{min} threshold, the number of iterations of the second phase is limited to O(1). See lines 1-2 in Algorithm 2.

Algorithm 2 Pseudo Code of Clustering Process

1. score $\longleftarrow \sum_{v=1}^{num\ of\ neighbors} \frac{1}{dist^2(me,v)}$
2. $\text{CH}_{prob} \longleftarrow$ MAX ($\text{C}_{prob} \times (E_{residunl}/E_{max}), P_{min}$)
3. *is_deterministic_CH* $\longleftarrow$ FALSE
4. **WHILE** $\text{CH}_{previous} \neq 1$ **DO**
5. **IF** ($\text{S}_{CH} \longleftarrow \{\upsilon$: υ is a candidate_CH or deterministic_CH$\}) \neq \varnothing$
6. **IF** ($\text{S}_{deterministic\text{-}CH} \longleftarrow (\{\upsilon$: υ is a deterministic_CH$\}) = \varnothing$
7. **IF** *NodelD = MOST_SCORE* (S_{CH}) **AND** $CH_{prod} = 1$
8. *CH_msg (NodelD, deterministic_CH, score)*
9. *is_deterministic_CH* $\longleftarrow$ *TRUE*
10. **ELSE-IF** $\text{CH}_{\text{prod}} = 1$
11. *CH_msg (NodelD, deterministic_CH, score)*
12. *is_deterministic_CH* $\longleftarrow$ *TRUE*
13. **ELSE-IF***Random(0,l)* < CH_{prod}
14. *CH_msg (NodelD, candidate_CH, score)*
15. $CH_{prvious} \longleftarrow CH_{prob}$
16. $CH_{prod} \longleftarrow$ MIN ($CH_{prod} \times$ 2, 1)
17. **END-WHILE**
18. **IF** *is_deterministic_CH = FALSE*
19. **IF** $S_{delermimstic\text{-}CH} \neq \varnothing$
20. *my_CH MOST_SCORE*($S_{delermimstic\text{-}CH}$)
21. *JOIN-CLUSTER* (*my_CH,NodeID*)
22. **ELSE**
23. *CH_msg (NodelD,delermimstic_CH,score)*

In the following, S_{CH} is defined as: $S_{CH} = S_{candidate\_CH} \cup S_{deterministic\_CH}$, where $S_{candidate_CH} = \{$All candidate CHs selected from iterations 1 to $i\}$ and $S_{deterministic_CH} = \{$All deterministic CHs selected from iterations 1 to $i\}$. During the execution of lines 3-23, a node may be a deterministic CH or a candidate CH or it may be covered by other nodes. In Algorithm 2, nodes with more energy have a better chance, CH_{prob}, of becoming a candidate CH. Whenever a node is selected as a candidate CH, it announces its new status by sending a message to all nodes within its cluster (Lines 13-14). In subsequent iterations, the node becomes a deterministic CH if CH_{prob} reaches one, and also the node has the highest score among the candidate CHs in its vicinity (Lines 5-12). Once selected, the node broadcasts a deterministic CH message to its neighbors within its cluster range. On the other hand, whenever a node receives a deterministic CH message from a cluster header, it can no longer become a CH. This regular node in its cluster radius selects one of the deterministic CHs based on the score of that deterministic CH. In Lines 15-16, each executor node doubles its CH_{prob} and goes to the next iteration of the loop. The run loop stops when $CH_{previous}$ reaches one. Accordingly, nodes with more energy execute loop commands earlier than others. This prevents the node with less energy from turning into a CH. Note that each candidate CH (or deterministic CH) node does not need to send a *CH_msg* message every time the loop is repeated. Except for the deterministic cluster headers identified so far, the other nodes make the final decision about their status by doing Lines 18-23. If a node receives at least one deterministic CH message, it selects the CH with the highest score from its adjacent deterministic CHs. In the event that a node does not receive any deterministic CH message while completing the loop execution, it finds itself uncovered

TABLE 1. Parameter setting.

Parameter	Value
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/signal
Idle power	13.5 mW
Sleep power	15 μW
Threshold distance (d_0)	75 m
Initial energy per node	2 J
Round time	20 sec
Round	5 frame
Data packet size	100 byte
Control packet size	25 byte
p_{min}	10^{-4}

and therefore has to present itself as a deterministic CH (Lines 22-23).

The execution time of the clustering algorithm depends on the number of times the loop instructions (Lines 4-17) are executed. This number also depends on the initial value of CH_{prob} which is limited by p_{min}. Since the maximum number of times the loop instructions are executed is fixed and does not depend on the number of nodes or other factors, the time complexity of the clustering algorithm is O(1).

IV. PERFORMANCE EVALUATION

In this section, similar to [29] and [30], the proposed algorithm is compared with LEACH, HEED, and EHEED algorithms. The simulations were performed in MATLAB software (similar to [8] and [31]). To compute the energy dissipation of radio hardware, a simple model [27] has been considered. Therefore, to transmit a k-bit message at distance d, the radio consumes: if $d > d_0$ then $E_{Tx}(k, d) = kE_{elec} + k\epsilon_{md}d^4$ otherwise $E_{Tx}(k, d) = kE_{elec} + k\epsilon_{fs}d^2$. To receive this message, the radio wastes: $E_{Rx}(k) = E_{Rx-elec}(k) = kE_{elec}$. Note that the value of d_0 is calculated as $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{md}}$. As studied in [20], this value for d_0 leads to the better results in the simulation of clustering algorithms.

The following common parameters and assumptions (similar to [8], [17], and [18]) are applied:

- To support different MAC protocols, each node has the necessary computing power and can perform signal processing functions.
- The sensor nodes are homogenous and quasi-stationary.
- Each node makes decisions based on local data and independently because the clustering is completely distributed. Also, the antennas equipped with GPS are not installed on the nodes.
- Table 1 shows the rest of the parameters.

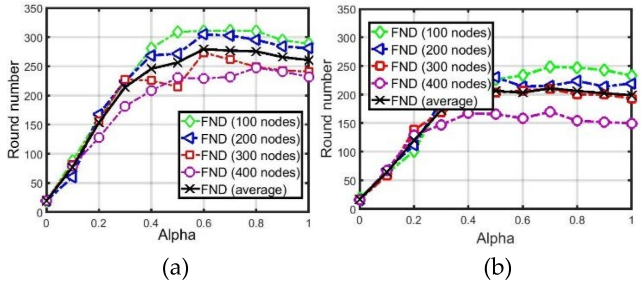

FIGURE 1. The lifetime of the proposed algorithm for the different number of nodes and alpha. (a) first scenario and (b) second scenario.

In the simulation experiment, the size of nodes 100, 200, 300, and 400 are chosen. Simulations are performed for two scenarios: (1) first scenario: A network with nodes randomly deployed in an area of 100 × 100 m^2 and the coordinate of BS is (50, 175). (2) second scenario: A network with nodes randomly deployed in an area of 200 × 200 m^2 and the coordinate of BS is (100, 275).

In HEED, EHEED, and MPCA, despite the difference in the number of the scenarios' nodes, by adjusting the cluster range, approximately five percent of the nodes are selected as CH. In the following, the appropriate value of α is studied, then the energy consumption and lifetime of the network, the number of clustering operations, and the number of clusters created in the network are investigated.

A. ALPHA PARAMETER

MPCA performs clustering partially. After a CH depletes a predetermined fraction of its energy (i.e. $E_{residual} \leq \alpha E_{CH}$), it directly informs its member nodes to perform reclustering at the beginning of the next round. Here, the value of α parameter is obtained by a manual trial and error process. To investigate the effect of α on network lifetime, MPCA was run for both scenarios (for this evaluation, the initial energy of the nodes was assumed to be 0.2 J). In Figure 1, each graph is the average of three runs. α varies from 0 to 1, and the number of nodes is 100, 200, 300, and 400. These charts are based on FND. Also, the average of the four mentioned plots is demonstrated.

When α is zero, reclustering is not performed during the lifetime of the network (that is, static clustering where clusters and CHs are fixed). In homogeneous networks where the nodes have similar capabilities, the energy of the CHs will run out quickly. If the CH dies, the cluster becomes inefficient. When α is one, similar to LEACH, HEED, and EHEED algorithms, complete clustering is performed in every round. The average graph in this figure shows that considering $\alpha = 0.7$, almost the best lifetime of the network is obtained. Therefore, the next evaluations (Figures 2-9) have applied this alpha value.

B. ENERGY CONSUMPTION

In this subsection, energy dissipation for clustering the network nodes and transmitting sensed data to BS are evaluated.

Note that, due to the high energy consumption of nodes by LEACH in scenario 2, the vertical axis of the figures does not have the same range of data.

Figure 2 evaluates the average energy loss per election for the simulated algorithms. Due to low clustering message complexity, the MPCA algorithm performs better than others. Another reason is that MPCA algorithm messages like HEED and EHEED are sent within the cluster radius. Since the LEACH algorithm does not apply the cluster radius to limit the message distribution range, the amount of energy consumed for clustering in this algorithm depends on the network diameter. This has caused the LEACH algorithm to have the highest energy dissipation. This issue is shown in the difference between the LEACH algorithm values in Figure 2(a) and Figure 2(b).

In Figure 3, the average energy loss of the clustering algorithms in each round is shown. Compared to the other two algorithms, the average of MPCA is much lower. The reasons for this difference are partial clustering, a low number of clustering messages, and not being done clustering at the beginning of every round.

Figure 4 illustrates the average energy dissipation for data transmission. Because MPCA uses the multi-criterion score to form better clusters, it expends less energy during the steady-state phase compared to other simulated algorithms. MPCA uses the multi-criterion score which is a combination of node centrality and node degree. Therefore, clusters formed by MPCA are better than HEED, EHEED, and LEACH. Figure 4(b) shows that LEACH consumes the most energy in data transmission to BS as well as intra-cluster communication. Since in LEACH, CHs may not be fairly distributed in the field, some cluster members may be located far from their respective CH. Therefore, its intra-cluster communication is not energy efficient.

To summarize Figures 3 and 4, Figure 5 evaluates the average total energy consumption per round for all compared algorithms. This figure shows that MPCA is an energy-efficient algorithm in which the size of the network diameter does not have a significant effect on its energy consumption.

The ratio of WSN clustering energy dissipation to total energy consumption (i.e., clustering overhead) for each algorithm is depicted in Figure 6. Compared to the HEED, EHEED, and LEACH algorithms, the MPCA algorithm creates much less clustering overhead in both scenarios. The result of the experiments in this section is that MPCA is an energy-efficient algorithm.

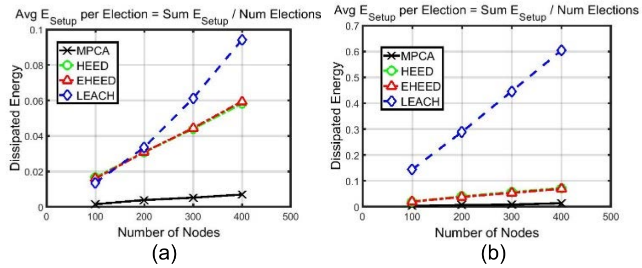

FIGURE 2. Average energy expended for clustering per election. (a) first scenario and (b) second scenario.

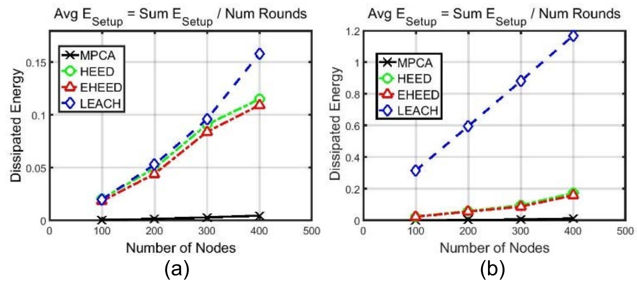

FIGURE 3. Average energy expended for clustering per round. (a) first scenario and (b) second scenario.

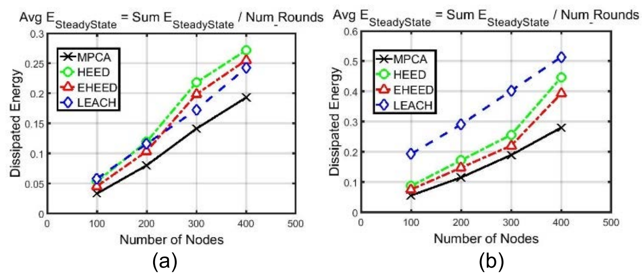

FIGURE 4. Average energy expended for steady-state in the network per round. (a) first scenario and (b) second scenario.

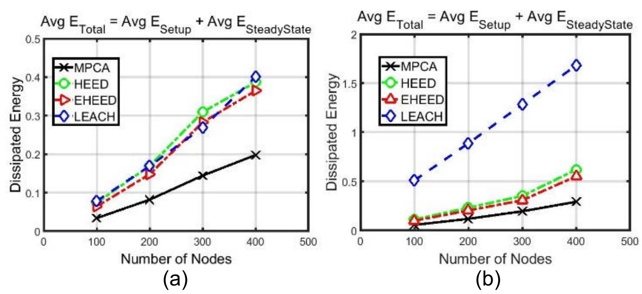

FIGURE 5. Total average energy expended in the network per round. (a) first scenario and (b) second scenario.

C. NETWORK LIFETIME

In Figure 7, for the different number of nodes, the network lifetime of the simulated algorithms with the two described scenarios are compared. In the MPCA, HEED, and EHEED algorithms, the FND is decreasing for 100-400 nodes. The reason is that to simplify the simulation, a simple power-aware routing algorithm is used (Figure 7(a) and (b)). This routing algorithm picks the least energy paths for data transmission through the CHs. Since the minimum energy paths are employed during the entire steady-state phase, the CHs of these paths (especially the cluster heads through which more paths pass) are depleted earlier than other nodes. To prolong the lifetime of the network, instead of the power-aware routing algorithm, probabilistic routing can be used in which different routes are used with various probabilities. In Figure 7, the superiority of MPCA over other simulated algorithms

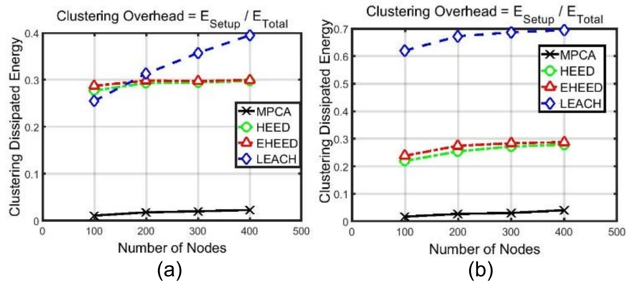

FIGURE 6. The overhead of clustering. (a) first scenario and (b) second scenario.

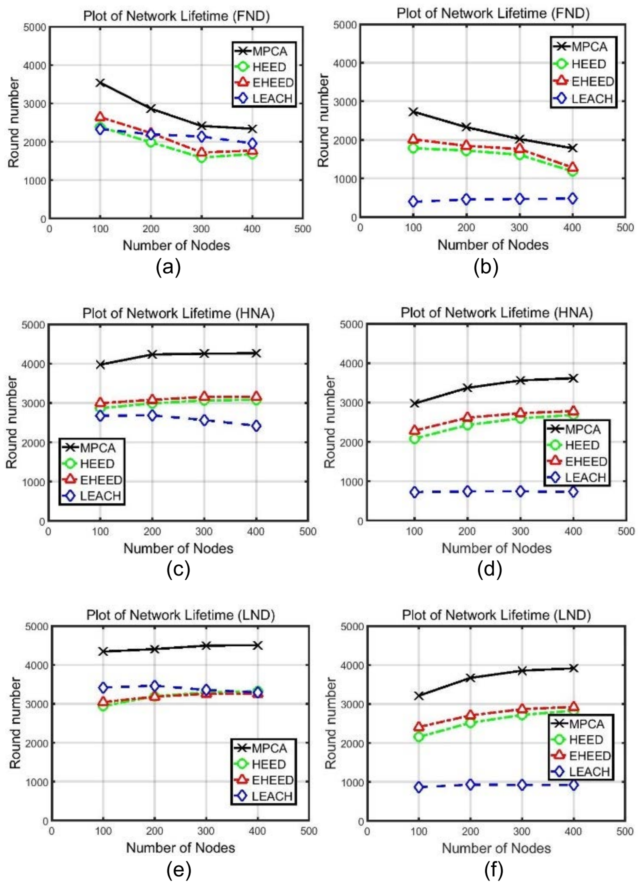

FIGURE 7. The lifetime comparison. FND in first (a) and second (b) scenarios. HNA in first (c) and second (d) scenarios. LND in first (e) and second (f) scenarios.

in every definition of the network lifetime (First Node Dies (FND), Half of the Nodes are Alive (HNA), and Last Node Dies (LND)) is illustrated. Consequently, in terms of the number of nodes and network size, MPCA is a scalable clustering algorithm.

D. THE NUMBER OF CH ELECTIONS AND CLUSTERS

Figure 8 compares the total number of reclustering (CH elections) performed until LND in the simulated algorithms. In HEED, EHEED, and LEACH algorithms, the setup phase is performed in every round. On the other hand, the total

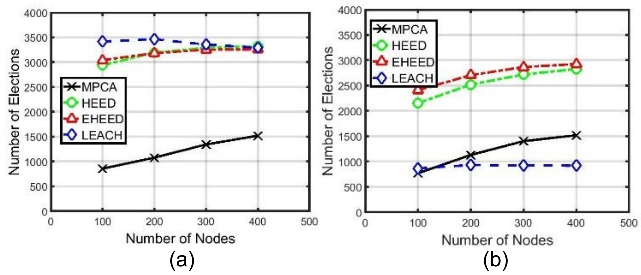

FIGURE 8. The number of clustering in LEACH, HEED, EHEED, and the proposed algorithm. (a) first scenario and (b) second scenario.

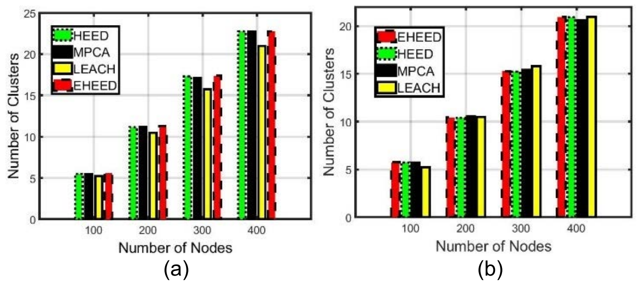

FIGURE 9. The number of clusters in LEACH, HEED, EHEED, and the proposed algorithm. (a) first scenario and (b) second scenario.

number of reclustering operations in MPCA is much less than in the other two algorithms. The figure shows that as the number of nodes increases, the number of reclustering also increases. This is because the LND value increases as the number of nodes increases (see Figures 7(e) and (f)). In the LEACH algorithm, because the LND value of scenario 2 is much lower than that of scenario 1, the number of cluster head elections in scenario 2 also reduces.

Fig. 9 demonstrates that these three algorithms have similar conditions in terms of the number of clusters. Therefore, according to Fig. 9 and Fig. 2-8, we can conduct that the novelty used in MPCA caused its superiority in terms of network lifetime and energy consumption in comparison with the other two algorithms.

V. CONCLUSION

In this paper, an energy-efficient distributed clustering algorithm called MPCA for WSNs is presented. The message overhead of the proposed clustering algorithm is low and also the cluster heads are fairly distributed throughout the network. This algorithm may be applied for applications that need scalability, a lengthened lifetime and the nodes are dispersals in a spacious field. It is assumed that the nodes of the network are quasi-stationary. Nodes are location-unaware and have equal importance. MPCA selects CHs based on their multi-criterion score and residual energy. On the other hand, each regular node becomes a member of the cluster head with the highest score in its vicinity. After clustering is completed, MPCA creates an inter-cluster multi-hop network over the CHs. To evaluate the efficiency of MPCA, well-known algorithms of LEACH, HEED, and EHEED were simulated. According to the simulation results, MPCA outperforms the compared algorithms in terms of network lifetime and energy saving. This advantage is due to (1) clustering being done on-demand and partially, (2) multi-criterion clustering being used, and (3) the number of messages required for clustering being low. While the MPCA algorithm clusters homogeneous nodes, the development of the algorithm to cluster heterogeneous nodes is left as future work.

REFERENCES

[1] C. Lin, G. Han, X. Qi, J. Du, T. Xu, and M. Martinez-Garcia, ''Energy-optimal data collection for unmanned aerial vehicle-aided industrial wireless sensor network-based agricultural monitoring system: A clustering compressed sampling approach,'' *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4411–4420, Jun. 2021.

[2] P.-Y. Kong and Y. Song, ''Artificial-neural-network-assisted sensor clustering for robust communication network in IoT-based electricity transmission line monitoring,'' *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16701–16713, Sep. 2022.

[3] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, ''A survey on mobility in wireless sensor networks,'' *Ad Hoc Netw.*, vol. 125, Feb. 2022, Art. no. 102726.

[4] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, ''Clustering objectives in wireless sensor networks: A survey and research direction analysis,'' *Comput. Netw.*, vol. 180, Oct. 2020, Art. no. 107376.

[5] F. Fanian and M. K. Rafsanjani, ''Cluster-based routing protocols in wireless sensor networks: A survey based on methodology,'' *J. Netw. Comput. Appl.*, vol. 142, pp. 111–142, Sep. 2019.

[6] A. S. Rostami, M. Badkoobe, F. Mohanna, H. Keshavarz, A. A. R. Hosseinabadi, and A. K. Sangaiah, ''Survey on clustering in heterogeneous and homogeneous wireless sensor networks,'' *J. Supercomput.*, vol. 74, no. 1, pp. 277–323, 2018.

[7] R. Dogra, S. Rani, H. Babbar, S. Verma, K. Verma, and J. J. P. C. Rodrigues, ''DCGCR: Dynamic clustering green communication routing for intelligent transportation systems,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16197–16205, Sep. 2022.

[8] P. Neamatollahi, M. Naghibzadeh, M.-H. Yaghmaee, and S. Abrishami, ''Distributed clustering-task scheduling for wireless sensor networks using dynamic hyper round policy,'' *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 334–347, Feb. 2018.

[9] J. Qin, Y. Zhu, and W. Fu, ''Distributed clustering algorithm in sensor networks via normalized information measures,'' *IEEE Trans. Signal Process.*, vol. 68, pp. 3266–3279, 2020.

[10] B. M. Sahoo, T. Amgoth, and H. M. Pandey, ''Particle swarm optimization based energy efficient clustering and sink mobility in heterogeneous wireless sensor network,'' *Ad Hoc Netw.*, vol. 106, Sep. 2020, Art. no. 102237.

[11] S. R. Pokhrel, S. Verma, S. Garg, A. K. Sharma, and J. Choi, ''An efficient clustering framework for massive sensor networking in industrial Internet of Things,'' *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4917–4924, Jul. 2021.

[12] S. Verma, S. Zeadally, S. Kaur, and A. K. Sharma, ''Intelligent and secure clustering in wireless sensor network (WSN)-based intelligent transportation systems,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13473–13481, Aug. 2022.

[13] I. Sohn, J.-H. Lee, and S. H. Lee, ''Low-energy adaptive clustering hierarchy using affinity propagation for wireless sensor networks,'' *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 558–561, Mar. 2016.

[14] D. Jia, H. Zhu, S. Zou, and P. Hu, ''Dynamic cluster head selection method for wireless sensor network,'' *IEEE Sensors J.*, vol. 16, no. 8, pp. 2746–2754, Apr. 2015.

[15] T. Kaur and J. Baek, ''A strategic deployment and cluster-header selection for wireless sensor networks,'' *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 1890–1897, Nov. 2009.

[16] X. Zhu, L. Shen, and T. S. P. Yum, ''Hausdorff clustering and minimum energy routing for wireless sensor networks,'' *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 990–997, Feb. 2009.

[17] I. Daanoune, B. Abdennaceur, and A. Ballouk, ''A comprehensive survey on LEACH-based clustering routing protocols in wireless sensor networks,'' *Ad Hoc Netw.*, vol. 114, Apr. 2021, Art. no. 102409.

[18] Z. Ullah, ''A survey on hybrid, energy efficient and distributed (HEED) based energy efficient clustering protocols for wireless sensor networks,'' *Wireless Pers. Commun.*, vol. 112, no. 4, pp. 2685–2713, Jun. 2020.

[19] S. A. Sert, A. Alchihabi, and A. Yazici, ''A two-tier distributed fuzzy logic based protocol for efficient data aggregation in multihop wireless sensor networks,'' *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3615–3629, Dec. 2018.

[20] S. A. Sert and A. Yazici, ''Increasing energy efficiency of rule-based fuzzy clustering algorithms using CLONALG-M for wireless sensor networks,'' *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107510.

[21] S. Arjunan and P. Sujatha, ''Lifetime maximization of wireless sensor network using fuzzy based unequal clustering and ACO based routing hybrid protocol,'' *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2229–2246, Aug. 2018.

[22] P. Chanak, I. Banerjee, and R. S. Sherratt, ''Energy-aware distributed routing algorithm to tolerate network failure in wireless sensor networks,'' *Ad Hoc Netw.*, vol. 56, no. 3, pp. 158–172, 2017.

[23] A. Cenedese, M. Luvisotto, and G. Michieletto, ''Distributed clustering strategies in industrial wireless sensor networks,'' *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 228–237, Feb. 2017.

[24] B. Baranidharan and B. Santhi, ''DUCF: Distributed load balancing unequal clustering in wireless sensor networks using fuzzy approach,'' *Appl. Soft Comput.*, vol. 40, pp. 495–506, Mar. 2016.

[25] X. Zhao and A. H. Sayed, ''Distributed clustering and learning over networks,'' *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3285–3300, Jul. 2015.

[26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, ''An application-specific protocol architecture for wireless microsensor networks,'' *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[27] O. Younis and S. Fahmy, ''HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,'' *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

[28] J. A. R. de Azua, A. Calveras, and A. Camps, ''Internet of satellites (IoSat): Analysis of network models and routing protocol requirements,'' *IEEE Access*, vol. 6, pp. 20390–20411, 2018.

[29] A. Zahedi, ''An efficient clustering method using weighting coefficients in homogeneous wireless sensor networks,'' *Alexandria Eng. J.*, vol. 57, no. 2, pp. 695–710, Jun. 2018.

[30] L. Abbad, A. Nacer, H. Abbad, M. T. Brahim, and N. Zioui, ''A weighted Markov-clustering routing protocol for optimizing energy use in wireless sensor networks,'' *Egyptian Informat. J.*, vol. 23, no. 3, pp. 483–497, Sep. 2022.

[31] V. Krishnaswamy and S. S. Manvi, ''Trusted node selection in clusters for underwater wireless acoustic sensor networks using fuzzy logic,'' *Phys. Commun.*, vol. 47, Aug. 2021, Art. no. 101388.

PEYMAN NEAMATOLLAHI received the B.S. and M.S. degrees in computer engineering, with a concentration in parallel and distributed systems, in 2007 and 2011, respectively, and the Ph.D. degree in computer engineering from the Ferdowsi University of Mashhad, in 2017. He is currently an Assistant Professor at Hakim Sabzevari University, Iran. He has published several journals and conference papers. His research interests include the Internet of Things (IoT), job scheduling, and distributed algorithms. He also serves as a Reviewer for the IEEE Transactions on Mobile Computing, IEEE Transactions on Vehicular Technologies, IEEE Sensors Journal, and some other journals.