Density Grid-Based Clustering for Wireless Sensors Networks

Manal Abdullah¹*, Hend Nour Eldin², Tahani Al-Moshadak³, Rawan Alshaik⁴, Inas Al-Anesi⁵
¹,²,³,⁴,⁵ Faculty of Computing and Information Technology, FCIT
King Abdulaziz University, KAU, Jeddah, SA.

Abstract

Wireless Sensor Networks WSNs are special networks consist of devices (sensor nodes) in large numbers and spatial distribution. They have various sensing capability and cooperate to accomplish common task. Clustering is one of the most effective techniques used to solve the problem of energy consumption in WSNs. Grid based clustering has proven its efficiency specially for high dynamic networks. The grids’ strategy used in this research is implemented on dense network and divides the network area into multiple grid cells with different densities (High, Low, and empty). Then grids are combined to form clusters as normal and advanced clusters. Cluster head is elected for each cluster based on high energy. This new suggested strategy is implemented and tested using MATLAB. The results show that this suggested strategy works well at 150 node WSN and grid size between 5-10 units where the network life time is 633 seconds approximately.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of Universal Society for Applied Research

Keywords: Grid-Based Clustering; Density-based Clustering; Clustering Algorithms; Wireless Sensor Network; Cluster head Election

* Corresponding author. Tel.: +966 509178668; fax: +0-000-000-0000.
E-mail address: maaabdullah@kau.edu.sa
1. Introduction

A wireless sensor network (WSN) is a group of spatially scattered hundreds or thousands sensor nodes that has the ability of sensing, communicating and computing. It embedded in physical spaces, continuously gather a big amount of data from the environment. Thereafter WSN is beneficial technology in many domains such as tracking, monitoring, scientific investigations and more.

Any WSN incorporate single or several sinks, single or several sources and many sensor nodes organized in an area with a certain topology. The sensor nodes can contain different sensors such as temperature sensor, sound sensor, pressure sensor, humidity sensor, etcetera. When this sensor nodes sense elements from the environment that it embedded in, the analogue signal is converted in to digital data by using the processing units inside the sensor nodes through the analogue to digital converter module, after that the data is send to the base station for processing. Wireless sensor node can communicate directly to base station also it can communicate with each other.

As shown in Fig. 1, sensor networks contain the following components:

- Data collecting: that has the ability of sensing and acquisition during transducers.
- Data transport: during the wireless/adhoc channels.
- Processing: that has the ability of analyzing data.

WSNs face many challenges in its design, where it is rigid to rules and policy changes such as user access and business operation. Any changes in business requirements will be hard to overcome by algorithms and it will need reprogramming or manual reconfiguration that is difficult to manage.

There is also many challenges comes from the intent of cost saving and the sensor node size where it has to be compact to suite the purpose. Being compact results in limited memory storage, limited computation strength and limited power source. In WSN the amount of data to be permanently stored into a data warehouse should be reduced by a certain summarization algorithm to accommodate the limited memory storage. Also because of the small size of the nodes, the node power is very important. While it is more practical to save the power and extend the life time of the network by using more efficient routing algorithm.

Clustering is the process of organizing objects into groups whose members are similar in some way. The cluster routing is an energy efficient protocol where the nodes will be grouped into a few groups and for every group one cluster head (CH) is elected. The data is collected and aggregated by CH from the nodes in the same cluster, and then it is transported to the base station. The cluster head is the only cluster node that is permitted to communicate with the base station. This will minimize the total used energy and minimize the congestion of the network.

In density-grid clustering algorithms, the data object is mapped first to a grid then these grids are clustered based on its density. Density-based clustering algorithms are remarkable not only to find arbitrarily shaped clusters but also to deal with noise in data. In density-based clustering algorithms, dense areas of objects are considered as clusters which are segregated by low-density area.

This paper focuses on cluster in a large-scale Wireless Sensor Networks in density area. This type of network contains a large number of densely deployed sensors in a certain area. This is like managing mobile nodes and improves the network resources efficiency in a density area. Moreover it inherits all the WSNs challenges that mentioned before (limited power capacity, Limited memory, etc.) The communication between the network nodes must be done in energy efficient manner. So, cluster routing techniques are used where the nodes will be grouped...
into a few groups, then the sensors will communicate only with cluster heads, subsequently cluster heads will communicate with the processing center.

In this research, grids strategy are implemented on dense network. We divide the network field into grids, where these grids are classified according to their density (High dense, Low dense, and empty), then the grids are aggregated to form clusters.

The rest of the research is organized as: Next section presents literature review. Section 3 explains the model of density grid-based clustering. Section 4 shows the suggested algorithms then the evaluation and experiment result is presented in section 5.

2. Literature Review

A Wireless Sensor Network is a collection of nodes organized into a network. Routing is an important technology in WSNs and can be divided into two categories:

1. Flat routing topology: where all nodes have identical functionality and carry out the same task in the network.
2. Hierarchical routing topology: where nodes implement different tasks in and are usually arranged into clusters.

The cluster has a leader node called cluster head CH and the other members are regular nodes. Cluster heads can be grouped further into hierarchical levels. CHs must have higher energy and perform data processing and information dissemination.

The importance of Cluster-based routing algorithms comes from a variety of advantages, such as greater scalability, less load, less energy consumption and more robustness.

Wireless sensors are used to gather data in different environments. The collected data is sent to processing center to determine characteristics of the environment or detect an event. The communication process must be done in low available energy.

So, any wireless sensor application pass three steps:

- Collect data by sensor.
- Sent data to base station.
- Generate decision based on analysis of collected data.

In WSN, data clustering concerns how to group a set of objects based on their similarity of attributes and/or their proximity in the vector space. Clustering algorithms are used in different fields such as marketing to find customers groups based on similarity of their behavior or cluster the products based on who purchase them, biology to classify plants and animals based on their features, cluster documents based on word similarity, cluster DNA based on edit distance, and in world wide web to discover groups of similar access patterns by clustering weblog data.

Clustering algorithms can be classified into different types based on cluster model such as hieratical clustering or connectivity based clustering. The most popular example is LEACH algorithm which is used a lot in WSN such as in. Abdulsalam and Kamel introduced W-LEACH which extends LEACH by selecting sensor to be CH if it has maximum p% of live time. The selection of CHs depends on weighted value wi which is assigned to each sensor Si. Then, clusters are formed by joining sensors to nearest CH. A ratio of x% of sensors in a cluster sent data to their CH. The sensors that are selected to be in x% are chosen based on their weights. Nurhayati introduced (ICRP) which is based on BCDCP and LEACH. This new hierarchical clustering algorithm can create equal clusters with same number of sensors like BCDCP method and then divided it into small cluster as in LEACH.

Centroid based clustering represents clustering based on a central vector which may not be a member of the cluster. This type uses k-mean algorithm where it finds k- cluster centers then assign objects to the nearest center. Jerusha, et al. modified the k-mean clustering algorithm by clustering the sensor nodes based on the shortest path distance and highest energy. Cluster mean is decided by using centroid method and CH is selected as the least distance between the cluster mean and cluster member.

Distributed based clustering depends on statistics based on distribution model. A well-known example of this type is Gaussian mixture model where objects are modeled with fixed size of Gaussian distributions which are initialized randomly and optimized the parameters iteratively to fit data set. Forero, et al. developed distributed clustering scheme for both probabilistic and deterministic approaches with unsupervised learning. They solved the centralized problem by using a distributed model which recasting the centralized problem to a set of smaller local clustering problems with consensus constraints on the cluster parameters.
Density based clustering creates clustering based on density where higher density area will be considered as cluster and spare area will be considered as being noise or bordering points. San-yang, et al. proposed density based cluster for WSN. The scale of each cluster is set based on density of the nodes with the network load and energy distribution balance.

3. Density Grid-Based Clustering Methodology

This research focuses on density-based and grid-based clustering methods. In grid-based clustering, the data space is partitioned into a finite number of equal space cells called grids, and then topological neighbor search is conducted on these grids to group the points of the closer grids. Grid-based clustering has fast processing time when it is compared with other types of clustering algorithms because all clustering operations are performed on the grid cells instead of the data objects. Many challenges are faced in this type of clustering which are:

- Determining the best size of grids: If the grid size is too large, more than one cluster could be formed inside one grid cell. In the other hand, if the grid size is small, a single cluster could be spread through more than one grid cell.
- The locality problem: When the data space has clusters with variable densities and arbitrary shapes, the global density threshold cannot result in clusters with less density.
- Selecting the merging condition that produces efficient clusters. Density-based clustering is another great method of efficient clustering. The idea of this type of clustering is about classifying the data set based on the dense regions, for example, if two points are very close to dense region, then these points considered to be elements of the cluster of the dense region. Density-based method has the ability to differentiate between the arbitrary shape clusters and to detect noise by considering dense areas of objects as clusters and low density area as noise.

Density grid-based algorithms work on three stages. In the first stage, partitioning the data space into equally space partitions called grids is performed. Then, each data point in data set is mapped into a grid. Finally, one of the density-based algorithms is applied into grids to form the clusters.

4. Density Grid-Based Clustering Algorithm

The proposed algorithm by this research combines the density-based and grid-based methods together to develop hybrid powerful clustering algorithm. It can discover arbitrary shape clusters, detect the outliers and noise, also it has a fast processing time, in which it only depends on the number of cells instead of the number of data objects. The algorithm goes through four stages as shown in Fig. 2. Create grids and map points, Classify grids, Create Clusters and Selecting CHs (pulling).

The following subsections detailed the algorithm stages:
4.1 Create Grids and Map Points

In this step the data space is partitioned into equally space partitions called grids. Then, each node is mapped into a grid by using algorithm 1. A grid is considered as a cells matrix where every cell contains a number of data points (nodes).

4.2 Classifying Grids

After the network space is divided, the number of data points in every non empty grid is calculated in order to determine the grid type (low or high dense grid). The standard deviation of the total number of nodes in the network is calculated according to equation (1). If the number of data points in a grid is more than or equal to the double of the standard deviation $\sigma$, the grid is considered as high dense grid. Thus the dense grid threshold is not need to set manually. (Algorithm 2)

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2} \quad (1)$$

Where $x_i$ is a single node in the network $\bar{x}$ is the mean of the network nodes, and $N$ is the total number of nodes in the space.

**Algorithm 1 Create grid**

1. Input: static data
2. Output: Grid matrix with data points
3. $\text{Gridx}=\text{ceil}(\text{Node x dimension} / \text{width})$
4. $\text{Gridy}=\text{ceil}(\text{Node y dimension} / \text{length})$
5. $\text{Grid Counter(grid x dimension, grid y dimension)} = \text{Grid Counter(grid x dimension, grid y dimension)} + 1$

**Algorithm 2 classify grids**

1. Input: Grid Matrix
2. Output: Grid type Matrix
3. Standard Deviation $= \text{Standard Deviation Function (Grid matrix)}$
4. threshold $= \text{Standard Deviation} \times 2$;
for every grid in the network field
if (there is no node in the Grid)
Grid type is 'empty';
else if (the number of nodes in the grid is less than threshold)
Grid type is 'Low';
else if (the number of nodes in the grid is more than or equal than threshold)
Grid type is 'High';
endif
endfor

4.3 Create Clusters

Clustering aims to minimize the overhead generated by the topology control and saves the node energy. In this proposed algorithm the nodes are clustered around the minimum high dense grid until certain number of nodes is reached as cluster threshold (algorithm 3). This is done according to the following steps as shown in Fig. 3:

Step 1. Searching for the minimum high dense grid. Grids at the borders are excluded. (algorithm 4)

Step 2. Examining the eight neighbors of the minimum high grid. If the neighbor grid is high or low dense, does not clustered before and the cluster threshold is not reached, then this grid will be included. (algorithm 5)

Step 3. Repeat step 2 satisfying the following conditions:
- If the neighbor is high dense and any of its adjacent neighbors are also high dense, then both are included in the cluster.
- If the neighbor is high dense and any of its adjacent neighbors are low dense, then both will be included in the cluster.
- If the neighbor is low dense and its neighbor is high dense, then both are included in the cluster.
- If the neighbor is low dense and its neighbor is also low dense, then both are included and this forms the border of the cluster.

Step 4. Repeat step 2 and step 3 for every next minimum high dense grid if it is not included in any cluster before.

Algorithm 3 Create clusters

Input: Grids in the field
Output: Clusters

1. %determine first high
2. for every grid not in the border of the network field
3. if (the grid type is dense)
4. number of High grid incremented ;
5. minimum = hold the x and y dimensions for the minimum high dense grid;
6. endif
7. endfor
8. minimum high dense grid x and y dimensions = findMinHigh(ClusterMatrix, width, length);
9. %Cluster around the minimum high
10. for (every High dense grids in center sub matrix)
11. if (Neighbour not in cluster && Neighbour Type is "High " && Cluster value less than or equal Cluster threshold)
12. Add Neighbour to cluster;
13. go to the next Neighbour( Grid);
14. elseif(Neighbour not in cluster && Neighbour Type is "Low " && Cluster value less than or equal Cluster threshold)
15. Add Neighbour to cluster;
16. go to the next Neighbour( Grid);
17. endif
18. endfor
4.4 Cluster Head Selection Second point

In this algorithm, after clusters are formed, the sensors would communicate only with cluster heads, subsequently cluster heads will communicate with the processing centre (base Station) This would save node energy as well as network energy.

In order to select a cluster heads (algorithm 6), the nodes in the network are divided into two types, normal nodes and advanced nodes with α times more energy than normal node. Advanced nodes have more opportunity to become a CH. Initially, the nearest advanced node from the base station is chosen as a cluster head. Then CH node in each cluster is elected according to the remaining energy.

Algorithm 4 findMinHigh function
Input: Cluster Matrix, width, high
Output: minimum high dense grid x and y dimensions
1. for (every grid in center sub matrix)
2.   if (Grid type is 'High' && it is not in a cluster
3.     && grid nodes< minimum )
4.     minimum = The Grid x and y dimension ;
5.     endif
6. endfor

Algorithm 5 add Neighbor of the Neighbor function
Input: Cluster Matrix
Output: Add Grid to Cluster
1. % add Neighbour of the Neighbour function
2. go to the next Neighbour (Grid) {
3.   while end not reached
4.     if (Neighbour not in cluster && Neighbour Type is "High " && next Neighbour Type is "High" && Cluster value less than or equal Cluster threshold)
5.     Add Neighbour to cluster ;
6.     go to the next Neighbour(Grid);
7.     elseif (Neighbour not in cluster && Neighbour Type is "High" && next Neighbour Type is "low" && Cluster value less than or equal Cluster threshold)
8.     Add Neighbour to cluster;
9.     go to the next Neighbour(Grid);
10.  elseif (Neighbour not in cluster && Neighbour Type is "Low "}
&&next Neighbour Type is "High" && Cluster value less than or equal Cluster threshold)
11. Add Neighbour r to cluster
12. go to the next Neighbour(Grid);
13. elseif(Neighbour not in cluster && Neighbour Type is "Low"
&& next Neighbour Type is "low" && Cluster value less than or equal Cluster threshold)
19. Add Neighbour to cluster;
20. break;
21. endif
22. endwhile
23. } end of go to the next Neighbour function

Algorithm 6 select CH
Input: Cluster Matrix, NumOfNodeInCluster
Output: Cluster head of cluster
1. for every cluster in Cluster Matrix
2. for every nod in a cluster
3. if (NodeEnergy >= m*numOfNodeInCluster+1)
4. node type is "Normal"
5. else
6. node type is "Advance"
7. endif
8. endfor
9. endfor
10. for every cluster in Cluster Matrix
11. for each advance node
12. select the nearest Advance node to base station to be "a cluster head"
13. endfor
14. endfor
15. // after that select the height energy advance node to be
16. //"a cluster head"

5. Experiments and Results

In this section the experimental evolution of density grid-based clustering algorithm is presented. The algorithms are implemented using MATLAB. All experiments were conducted on a 2.40 GHz machine with 8 GB memory, running on Windows 8 Operating system.

5.1 Data Sets and Evaluation

To evaluate proposed clustering algorithm quality, static non homogenous distributed data sets are used in order to create synthetic data set. The network field is 100*100 dimensions and has 100 nodes. It is divided into multiple grids with 10*10 dimensions. The maximum data packet length that the node can send is 6400byte. The initial energy for every node is 0.5 Joule and the maximum number of sensor nodes per cluster is 20 as in 23.

To disperse between low and high dense grids, we calculate the number of nodes in every non-empty grid. The standard deviation of the total network nodes is calculated. If the number of data points in a grid is more than or equal the standard deviation, the grid is considered as a high dense grid. Thus the dense grid threshold is not need to set manually.
In the first stage of this algorithm we determine the grid type depending on the parameters mentioned above. Then nodes are colored depending on the grid type as shown in Fig. 4. Blue color is for dense grid and red color is for red color as shown in Fig. 4. Based on the above parameters, Fig. 5 showed the formed eight clusters for 100 sensor nodes in the field this to this point forward.

**Fig. 4. Coloring Nodes Depending on Grid Type.**

**Fig. 5. Formed Clusters.**

### 5.2 Experimental Results

The algorithm run for different grid sizes to show the effect of grid size on the computational time, number of formed clusters, and network lifetime. Table 1 shows the computation time in seconds for different grid size range from 1 to 15 units. For this range, number of formed clusters range from 90 to 6 clusters. Fig. 6 represents grid different grid size with the number of formed clusters. Network lifetime is an important consideration of WSN. Fig. 7 represents the network lifetime versus grid size. From the figure, network lifetime range between 230 to 250 seconds between grid sizes 10 to 15. Although networks with 1 grid size have most better performance (life time) than other grid sizes that is 900 seconds, this experiment is excluded because each cluster has average of 1 to 2 nodes maximum which violates the logic of cluster forming strategy. In addition, its computation time to form clusters is very high as shown in Table 1.

**Fig. 6. Number of Formed Clusters for Different Grid Sizes.**
Fig. 7. Network Lifetime for Different Grid Sizes.

Table 1. Computation Time for Different Grid Sizes.

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Computation Time/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.098</td>
</tr>
<tr>
<td>5</td>
<td>0.6488</td>
</tr>
<tr>
<td>10</td>
<td>0.55667</td>
</tr>
<tr>
<td>15</td>
<td>0.2863</td>
</tr>
</tbody>
</table>

The effect of increasing the number of nodes in WSN field is an important results for our experiments, we tried to test the clustering method on higher number of nodes range between 100 nodes to 300 nodes. We choose the grid size of 10*10 dimensions because it shows the most suitable results in terms of network life time and computation time. The results are shown in Fig. 8 and Fig. 9. The computation time of the experiments are shown in Table 2. The number of formed clusters ranges from 8 to 25 clusters while the network lifetime ranges between 100 to 630 seconds approximately. This clustering method achieves the best performance when the number of nodes is 150 nodes in an area of 100*100 because the network life time has the highest value as 633.0864 seconds.

Fig. 8. Number of Formed Clusters for Different Number of Nodes.
6. Conclusions and Future Work

In this paper, we proposed a new clustering method for wireless sensor networks WSNs based on of density grid-based clustering. In this method, the network area divided into grids which are classified as high dense, low dense and empty grids according to the number of nodes in the cluster. These grids combined to form clusters where empty grids are excluded, two adjacent high dense grids are joined in the cluster, two adjacent high dense grid and low dense grid are also joined in the cluster, and two adjacent low dense grids will become as outlier of a cluster. To determine an appropriate cluster head, cluster nodes are distributed as normal nodes and advance nodes where the cluster head initially is chosen from the advance nodes with minimum distance to base station. Then the cluster head will be elected based on highest energy. This new suggested method is implemented and tested using MATLAB. Experiments showed that to determine the appropriate grid size then the number of nodes for each cluster, we conducted the experiments for approximately 10 times where average results are considered. The most appropriate grid size which reveals best results in terms of network life time is between 5 to 10 units with 150 nodes in the WSN area.

As a future work it is important to compare the results with one of the famous clustering algorithms for WSNs such as LEACH protocol or any of its modifications to validate the results. Other performance metrics such as delay could also be considered.
References


