



**HAL**  
open science

# Genetic Centralized Dynamic Clustering in Wireless Sensor Networks

Mekkaoui Kheireddine, Rahmoun Abdellatif, Gianluigi Ferrari

► **To cite this version:**

Mekkaoui Kheireddine, Rahmoun Abdellatif, Gianluigi Ferrari. Genetic Centralized Dynamic Clustering in Wireless Sensor Networks. 5th International Conference on Computer Science and Its Applications (CIIA), May 2015, Saida, Algeria. pp.503-511, 10.1007/978-3-319-19578-0\_41 . hal-01789981

**HAL Id: hal-01789981**

**<https://inria.hal.science/hal-01789981>**

Submitted on 11 May 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Genetic Centralized Dynamic Clustering in Wireless Sensor Networks

Mekkaoui Kheireddine, Rahmoun Abdellatif, Gianluigi Ferrari

GeCoDe Laboratory, University of Dr Tahar Moulay, Saida, Algeria  
EEDIS Laboratory, University of Djillai Liabes, SBA, Algeria  
WASNLab Laboratory, University of Parma, Parma, Italy

**Abstract.** In order to overcome the energy loss involved by communications in wireless sensor networks (WSN), the use of clustering has proven to be effective. In this paper, we proposed a dynamic centralized genetic algorithm (GA)-based clustering approach to optimize the clustering configuration (cluster heads and cluster members) to limit node energy consumption. The obtained simulation results show that the proposed technique overcomes the LEACH clustering algorithm.

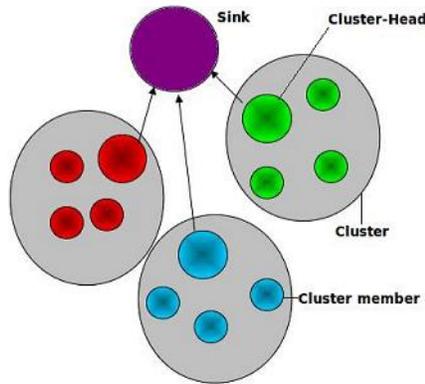
## 1 Introduction

Wireless sensor networks (WSNs) are used in many domains, such as military surveillance, disaster management, forest fire detection, seismic detection, habitat monitoring, biomedical health monitoring, inventory tracking, animal tracking, hazardous environment sensing and smart spaces, general engineering, commercial applications, home applications, underwater applications, etc [1]. Indeed, according to [2], WSNs are considered to be one of the new technologies that will change our life, they are listed, also in [3], as one of the key technologies of the internet of things.

The sensor nodes (or motes) are physical entities characterized by: (i) a battery with a limited energy; (ii) a processor with limited processing capabilities; (iii) and a transceiver [4]. The nodes can be deployed in monitoring areas in order to gather multiple types of information (e.g., humidity, light, temperature, wind,...) and then transmit the gathered information to the gateway sensor node (Access Point or Sink), possibly using multi-hop routing strategy [5]. In turn, the sink transmits the collected information to the end users.

Since it might often be difficult to replace exhausted batteries (e.g., WSNs may be deployed in inaccessible areas) [6], extending the lifetime of the WSN is crucial. In the literature, many papers show that the source of highest energy consumption in the sensor node is the transceiver [7], making strategies which minimize the use of the transceiver very attractive. Several techniques can be used to save energy, among which clustering consists in grouping sensors in several clusters, so that each cluster has a single cluster-head and several cluster-members. In each cluster, the cluster-members gather information on the sensed area and send it to the cluster-head. In turns, the cluster-head processes the data

received from its members and sends it to the sink. A graphical representation of a clustered WSN is shown in Fig. 1.



**Fig. 1.** Clustered WSN.

In a clustered WSN, data collected by the sensors is communicated to its cluster-head, for data processing and redundancy elimination. Therefore, sensors communicate data over short distances in each cluster (to cluster-heads), so that the energy spent in communication will be lower than that spent with sensors communicating directly to the sink [8].

Clustering can be static or dynamic. In a *static* scenario, the cluster-heads are fixed and tend to exhaust their energies rapidly, making this clustering unsuitable for WSNs [9]. In fact, the network becomes nonfunctional in the absence of cluster-heads. In the presence of *dynamic* clustering, the clusters change over the time, equalizing the energy consumption across all nodes and, thus, extending the network lifetime.

In this paper, our goal is to maximize network lifetime (defined as the time interval from the nodes' deployment to the instant at which a given percentage of deployed nodes die [6]) by minimizing the average energy consumption of all nodes. In order to do this, all nodes can be promoted to the role of cluster-heads. In order to reach this goal and guarantee full coverage (i.e., the clusters are spread over the entire network), we rely on the use of a genetic algorithm (GA), which determines, in each cycle, whether or not a node can be chosen to play the role of a cluster-head.

## 2 Related Work

The idea of using clustering has been adopted by many authors. The linked clustering algorithm (LCA) was one of the first approaches [10]. In the LCA algorithm, each node has a unique ID. In this algorithm a node play the role of cluster-head if its ID is the highest one in its neighboring.

LEACH is the most popular clustering algorithm for WSNs [9]. LEACH allows a fixed percentage of nodes to become cluster-heads (namely, 5% of the nodes) and leads to the creation of clusters in a distributed way, with the nodes taking autonomous decisions. Each node decides to become a cluster-head with probability  $p$ . A node which does not become cluster-heads determines its cluster by choosing the nearest cluster-head. On average, LEACH provides low energy consumption and a uniform energy distribution among the nodes. However LEACH has also some drawbacks. Because of the probabilistic selection of the cluster-heads, a node with a very low energy can be selected as cluster-head. Moreover, since the selection of cluster-heads is probabilistic, the chosen cluster-heads may be placed in the same area, so that a good coverage can not be guaranteed: in fact, some nodes will be disconnected from the network (i.e., they will not attach to any cluster-head). Moreover, the use of a fixed percentage of cluster-heads may lead (network-wide) to higher energy consumption, as the number of cluster-heads depend on several factors, such as node spatial density [11].

EEHC is a randomized and distributed clustering algorithm, whose goal is to maximize the network lifetime [12]. This algorithm is executed in two levels. In the first level, denoted as “initial,” volunteer nodes, which do not belong to any cluster, may decide to be cluster-heads with probability  $p$  and they announce their decisions to their neighbors. The nodes that do not receive an announcement, within a specified time interval  $t$ , become forced cluster-heads. In the second level, denoted as “extended,” the clustering algorithm is recursively repeated to form hierarchical clustering, where new cluster-heads are selected from the already formed cluster-heads, until a final base station is reached.

In [13], the authors consider a GA and adapt, on the basis of software services, its parameters to determine the energy consumption and, therefore, extend the network lifetime. In [6], the authors proposed a GA-inspired routing protocol (GROUP): in particular, they use GA and simulated annealing (instead of the greedy chain) to select routing paths efficiently.

## 3 System Model

The conditions and assumptions behind the considered network model are compliant with those considered in [9] for LEACH. More precisely, they can be summarized as follows.

1. The base station is fixed, is not energy-constrained, and has a high computing capacity.
2. All the nodes deployed are energy/power-constrained and homogeneous.

3. The data processing power is very low with respect to the power required to transmit and receive data.

The nodes' radio communication specifications are set as in [9, 4, 6]. In particular, we assume that the radio module dissipates:  $E_{elec} = 50$  nJ/bit in transmission/receiver circuitry; and  $\epsilon_{amp} = 100$  pJ/bit/m<sup>2</sup> in the transmitter amplifier. Considering free space communications, in order to transmit a  $k$ -bit message over a distance  $d$  (dimension: [m]) a node consumes the following amount of energy:

$$E_{Tx}(k, d) = E_{elec} \times k + \epsilon_{amp} \times k \times d^2. \quad (1)$$

When receiving a  $k$ -bit message a node consumes the following amount of energy:

$$E_{Rx}(k) = E_{elec} \times k. \quad (2)$$

## 4 The Proposed Approach

### 4.1 The Problem

In a clustered WSN, if a few cluster-heads are used, then most of the nodes are likely to have a long transmit radio range to send the collected data to their cluster-heads and this tends to quickly deplete their batteries' energies. If a large number of cluster-heads is used, this leads mostly to a one-hop network (most nodes are cluster-heads and must reach the base station in one hop): this consumes also quickly the battery energy [4, 8].

The best clustering strategy consists in optimizing (i) the number of cluster-heads and (ii) their positions. In particular, a node can be promoted to cluster-head according to several parameters: its residual energy, its distance to the sink, and the sum of the distances to its cluster-members. This suggests the use of GAs to find the optimal combination of these parameters.

### 4.2 The Proposed Algorithm

In this paper, we consider *dynamic* clustering, i.e., re-clustering is considered to avoid early death of cluster-heads. The proposed GA is executed at the sink (i.e., it is *centralized*), due to the needed computing capacity, and the obtained results (in terms of clustering configuration) are communicated to the nodes. At each re-clustering round, each node can then be either a cluster-head or a cluster-member. This centralized approach is expected to overcome the main limitations of LEACH, where the number of cluster-heads is fixed and their spatial distribution is arbitrary, i.e., there is no coordination [9].

In order to use a GA, a WSN needs to be "codified." In particular, we use a binary representation, in which each node is represented either by 0 (if it is a cluster-member) or by 1 (if it is a cluster-head). Each codified network is called a "chromosome." A set of chromosomes is called a "generation."

The used GA is based on exploration and exploitation of the entire research space using an evolutionary strategy, it helps us to find an optimal combination of cluster-heads, cluster-members and their distributions in the monitoring area, among many combinations existing in the research space, making the energy consumption and the network coverage, optimal. Each potential solution is characterized by a value called fitness, which determines the optimality of solutions. In correspondence to a generation, the GA keeps the best chromosomes and drops others according to their fitness function. Each chromosome, in fact, represents a potential solution. The GA then applies the following genetic operators to generate new offsprings [14].

- *Selection*. The selection process is used to choose the best chromosomes from a generation. In our simulation, the roulette wheel algorithm is used to perform the selection.
- *Crossover*. To apply crossover, we choose arbitrary two chromosomes from a generation, we choose, also, two random positions in the chosen chromosomes and we used the two point crossover, to generate two new offsprings, that will belong to the next generation.
- *Mutation*. The mutation is used to avoid the super chromosome problem. It means if one chromosome is selected many times in the same generation, the crossover will not produce new chromosomes, since the parents are the same chromosome. Hence, the mutation is used to change, in each chromosome, an arbitrary bit. Several tentatives have been performed so to come up with the best-run GA parameters in terms of runtime and convergence. The best crossover and mutation probabilities are 0.75 and 0.2, respectively.

As mentioned above, each chromosome is then evaluated with a fitness function which attributes a higher chance to the best solutions to survive. The fitness of a candidate chromosome can be expressed as follows:

$$\text{Fitness} = f(NNN, NCH, DNCH, RECH) \quad (3)$$

where:

- $NNN$  is the number of networked nodes;
- $NCH$  is the number of cluster-heads;
- $DNCH$  is the sum of the distances between the cluster-members (CMs) and their cluster-heads (CHs), i.e.,

$$DNCH = \sum_{i \in \{\text{CHs}\}} \sum_{j \in \{\text{CMs}\}} \text{Distance}(\text{CH}_i, \text{CM}_j);$$

- $RECH$  is the sum of residual (cumulative) energy at the cluster-heads (dimension: [mW]), i.e.,

$$RECH = \sum_{i \in \{\text{CHs}\}} \text{Residual energy at the CH}_i.$$

In order to optimize the proposed clustering mechanism, we consider the following (heuristic) fitness function:

$$\text{Fitness} = (NNN)^{\alpha_1} + \left(\frac{NNN}{NCH}\right)^{\alpha_2} + (10^3 \cdot RECH)^{\alpha_3} + DNCH^{\alpha_4}$$

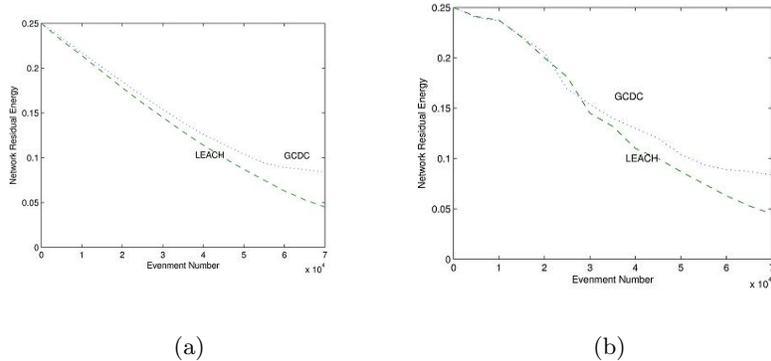
where: the exponential parameters  $\{\alpha_i\}_{i=1}^4$  need to be properly optimized; the fraction  $NNN/NCH$  represents the average cluster dimension; the multiplicative term  $10^3$  used for  $RECH$  properly weighs the energy dimension. By trial and error, the best fitness function (i.e., the best configuration of the exponents  $\{\alpha_i\}_{i=1}^4$ ) turns out to be

$$\text{Fitness} = (NNN)^6 + \left(\frac{NNN}{NCH}\right)^5 + (10^3 \cdot RECH)^2 + DNCH.$$

## 5 Performance Analysis

In order to validate the proposed clustering approach, we carry out a simulation-based performance analysis, considering different scenarios, by varying the node spatial density, the number of nodes, and the sink position. In each scenario, a given number of sensors is randomly deployed in a square monitored area, with side length  $800 \times 600$ . The sink, placed within the region, runs the GCDC algorithm (*for Genetic Centralized Dynamic Clustering*) and informs the sensors of the decided clustered configuration. After receiving the decision, each node knows if it is a CH or a cluster member. The GCDC algorithm is periodically run by the sink in order to avoid that a node death compromises network connectivity. We assume that all nodes have batteries with initial energy equal to 0.25 J. The dimension  $k$  of the messages to be transmitted is set to 100 bits. We assume that random “events” (e.g., acoustic signal detection, motion detection, etc.) happen in the monitored area: in particular, each random event is detected by its nearest neighbor, which needs to report this observation to the sink. In all considered scenarios, the performance of the GDC algorithm is compared with that of LEACH. Two values of the initial number of nodes in the WSN are considered: 100 (low node spatial density) and 1000 (high node spatial density).

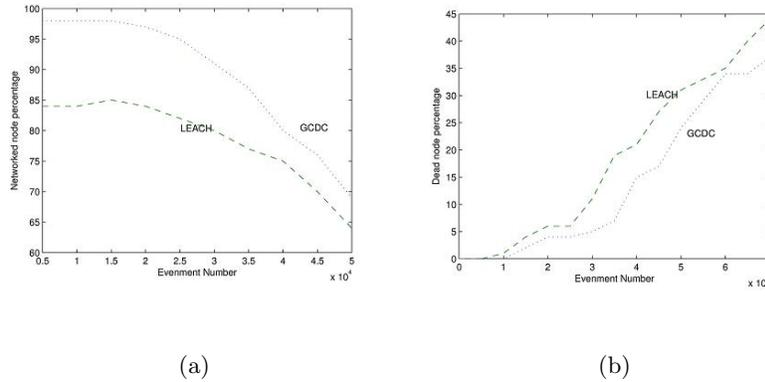
In Figure 2, the residual network energy is shown as a function of the simulation time (expressed in event number), considering two values for the initial number of nodes: (a) 100 and (b) 1000. It can be observed that GCDC algorithm allows to save more energy than LEACH. The energy saving is not relevant at the beginning, whereas it becomes more significant as the time passes by. This is due to the fact that the GCDC algorithm updates the network clustered topology very efficiently. This behaviour is more pronounced in the scenario with 100 nodes (low node spatial density).



**Fig. 2.** Network residual energy as a function of the simulation time (in terms of event number). The initial number of nodes in the network is set to: (a) 100 or (b) 1000.

In Figure 3, we investigate the network connectivity evolution, considering (a)  $NNN$  (i.e., the network coverage) and (b) the number of dead nodes, as a function of the simulation time (in terms of event number). In both cases, the initial number of nodes in the WSN is set to 100 (low node spatial density). From the results in Figure 3 (a), it can be observed that the number of nodes connected (i.e., becoming cluster members or heads) by the GCDC algorithm is larger than that guaranteed by LEACH. This is more evident at the beginning of the simulation, when all nodes (having full battery energies) could be connected, whereas the improvement brought by GCDC reduces for advancing simulation time, as a larger and larger number of nodes die. The performance difference is due to the fact that LEACH *a priori* sets the number of CHs to 5% of the total number of nodes without identifying their positions: this likely leads to overlapped clusters (i.e., two CHs may be close to each other), leaving other nodes (without a sufficiently close CH) disconnected. The GCDC algorithm does not determine a priori the number of CHs but, rather, the GA determines the optimized number of CHs, along with their positions, to cover the entire monitored area efficiently. In Figure 3 (b), the number of dead nodes (after energy depletion) is shown: as expected from Figure 3 (a), the death rate with GCDC is lower than that with LEACH, owing to the clustering procedure which takes into account the nodes' residual energies.

In Figure 4, the network connectivity evolution, considering (a)  $NNN$  (i.e., the network coverage) and (b) the number of dead nodes as functions of the simulation time (in terms of event number), is investigated in a scenario with 1000 initial nodes (high node spatial density). By comparing the results in Figure 4 (a) with those in Figure 3 (a), it can be concluded that the performance improvement, in terms of  $NNN$ , brought by GCDC is more pronounced in dense network. In particular, since, according to the results in Figure 4 (b), the death rates of GCDC and LEACH are approximately the same, it means that the



**Fig. 3.** Network connectivity evolution, in terms of (a)  $NNN$  and (b) number of dead nodes, as a function of the simulation time (in terms of event number). The initial number of nodes in the WSN is 100.

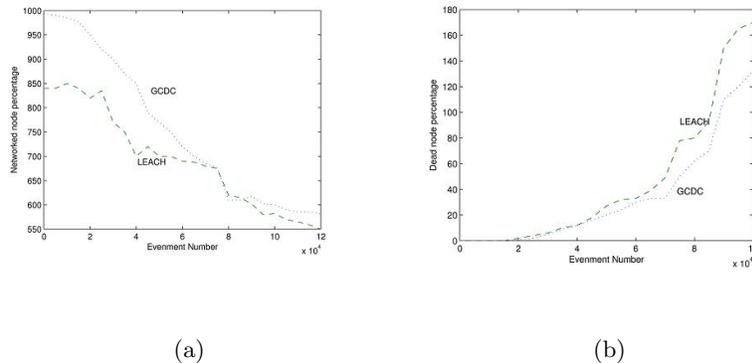
GCDC is very efficient in reclustering the topology in order to guarantee a high level of connectivity to the surviving nodes.

## 6 Conclusion

In this paper, we have presented a novel clustering algorithm, denoted as GCDC, which uses a GA to optimize the number and the corresponding locations of CHs. The performance of our algorithm has been compared with that of LEACH. The obtained results show that the proposed clustering algorithm reduces the (network-wide) energy depletion rate and guarantees a better network coverage. An interesting research direction consists in applying the proposed GA-based clustering algorithm to duty-cycled WSNs.

## References

1. Huafeng W. and al, An acoa-afsa fusion routing algorithm for underwater wireless sensor network, *International Journal of Distributed Sensor Networks*, pp.4110-4118, 2012.
2. Ilyas, M. and Mahgoub, I. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, CRC Press LCC, 2012.
3. Jianbin, X., Ting Z., Yan Y., Wenhua W. and Songbai L. (2013), Cooperation-based ant-colony algorithm in wsn, *Journal of Networks*, vol. 8, no. 4,
4. Mekkaoui k. and Rahmoun A., Short-hops vs. long-hops - energy efficiency analysis in wireless sensor networks, in *CIIA 2011: Proceedings of the Third International Conference on Computer Science and its Applications (CIIA11)*, University of Saida, Algeria, pp. 13-15.
5. Akyildiz I. F. and Vuran, M. C., *Wireless Sensor Networks*, Jon S. Wilson.



**Fig. 4.** Network connectivity evolution, in terms of (a)  $NNN$  and (b) number of dead nodes, as a function of the simulation time (in terms of event number). The initial number of nodes in the WSN is 1000.

6. Chakraborty A., Mitra S. K. and Naskar M. K. (2011), A genetic algorithm inspired routing protocol for wireless sensor networks, *International Journal of Computational Intelligence Theory and practice*, vol. 6, No. 1, pp.
7. Odey A. J. and Li D. (2012), Low power transceiver design parameters for wireless sensor networks, *Wireless Sensor Network*, vol. 4, No. 10, pp.243-249.
8. Abbasi, A. A. and Younis, M. (2007), A survey on clustering algorithms for wireless sensor networks, *Computer communications*, vol. 30, No 14, p. 2826–2841.
9. Heinzelman, W. R., Chandrakasan, A. and Balakrishnan, H. (2000), Energy efficient communication protocol for wireless microsensor networks, in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, ser. HICSS 00. Washington, DC, USA: IEEE Computer Society, pp. 8020-.
10. Dechene, D. J., Jardali, A. E., Luccini, M. and Sauer A. (2006), *Wireless sensor networks a survey of clustering algorithms for wireless sensor networks*, Department of Electrical and Computer Engineering, The University of Western Ontario, Canada, Tech. Rep.
11. Jin, S., Zhou, M. and Wu, A. S.(2003), Sensor network optimization using genetic algorithm, in *Proceedings of the 7th World Multiconference on Systemics, Cybernetics, and Informatics*, Orlando, FL, pp. 109–116.
12. Bandyopadhyay, S. and Coyle, E. J. (2003), An energy efficient hierarchical clustering algorithm for wireless sensor networks, in *INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, pp. 1713-1723.
13. Norouzi, A., Babamir, F. S. and Zaim, A. H. (2011), 'A New Clustering Protocol for Wireless Sensor Networks Using Genetic Algorithm Approach, *Wireless Sensor Network*, vol. 3, No.11. pp.362-370
14. Holland, J. H. (1992), Genetic algorithms, *Scientific american*, vol. 267, No. 1, pp.66–72.