International Journal of Innovative Computing, Information and Control
Volume **16**, Number **5**, October **2020**

 ISSN 1349-4198
pp. 1623–1638

CUSTOMIZABLE HIERARCHICAL WIRELESS SENSOR NETWORKS BASED ON GENETIC ALGORITHM

Elma Zanaj[1], Ennio Gambi[2], Blerina Zanaj[3] and Deivis Disha[2]

[1]Department of Electronics and Telecommunications
Polytechnic University of Tirana
"Nene Tereza" Square No. 1, Tirana 1005, Albania
ezanaj@fti.edu.al

[2]Department of Information Engineering
Polytechnic University of Marche
Via Brecce Bianche, 12, Ancona 60131, Italy
e.gambi@staff.univpm.it; d.disha@pm.univpm.it

[3]Department of Mathematics and Informatics
Agricultural University of Tirana
Kodër Kamëz, SH1, Tirana 1000, Albania
bzanaj@ubt.edu.al

Received May 2020; revised August 2020

Abstract. *This paper presents a design tool for WSNs based to optimization mechanisms that enable the user to build a network topology selected by optimizing different potential configurations. Through the suggested mechanisms, it can also be determined the optimal number of active network nodes to meet the requirements of a specific application before the physical implementation of the system. In this paper we propose a customizable heuristic approach of WSNs topology design based on genetic algorithms, in order to take account of specific application parameters: coverage, energy efficiency, system node degree in the environment and network lifetime. Innovation of our method is in a proper representation of different weight coefficients used to study the optimization of the network. The results of simulations prove that the overall network performance of the proposed heuristic deployment approach is superior compared to random WSN deployment.*
Keywords: Genetic algorithm, Weight, Optimization, Fitness function

1. **Introduction.** A Wireless Sensor Network (WSN) consists of a relatively large number of sensor nodes organized in a network in a certain area with the primary objective to autonomously perform specific tasks, such as event detection, physical parameters measurements and target object tracking. Technological advancement in electronics related fields, especially developments in the embedded systems has made it possible to increase the reliability, capacities, efficiency and to decrease the size and the cost of sensor nodes [1]. The utilization of the WSNs' advantages like the dynamic self-organizing features and decentralized functionality through wireless communication has dramatically increased the application of WSNs in many diverse fields and the list is still growing. The largest groups are the commercial applications, which include using of IoT in cities [2], safety systems [3], healthcare detection systems [4] or health monitoring through wearable sensor [5] and environmental monitoring systems [6]. A wide range of applications with an even wider range of performance requirements has resulted in development of variety of protocols which include several variable parameters [7]. WSN differs from traditional

DOI: 10.24507/ijicic.16.05.1623

wireless networks in many ways due to their nodes with limited capacities, strict energy constraints and application-specific characteristics. A typical WSN is a homogenous network, composed by a collection of sensor nodes which collaborate with each other in order to fulfill a certain task. The data collected from the network may be of different types due to various application scenarios; thus different types of applications have their own specific requirements. These requirements are converted into specific design properties of a WSN. QoS's parameters are generally dependent on the task assigned. Packet delivery ratio in WSNs is no longer a sufficient evaluation metric; instead, different applications may take their own requirements into consideration. A certain network configuration is unlikely to meet the performance criteria of every possible application. The traditional protocol stack TCP/IP, which has been successfully applied to traditional networks may not be suitable for WSN. Therefore, many new communication algorithms have been developed for different applications.

So, several studies are developed for different clustering methods, when hierarchical architectures have taken in consideration of what a cluster can be thought. These approaches aim to provide more accurate clustering by reducing the clusters steps combination. So, Nazari et al. [8] presented a new bottom-up hierarchical clustering algorithm that uses intersection point as linkage criterion. This approach provides more accurate clustering result since none of nearest neighbors of a data point can be missed.

Considering the impact of the network organization on the overall system performance, WSN topology design is an important matter prior to network deployment. The sensor nodes are usually deployed deterministically or heuristically in controlled environments such as homes, factories, residential buildings or hospitals. Alternatively, they are randomly deployed in uncontrolled environment, referring to hazardous and dangerous regions such as battlefields, toxic regions and areas impacted by natural disasters.

This is why some studies are focused in multi-level clustering algorithms as methods for optimization of data acquisition. So, they use Ant Colony Optimization (ACO) [9] or multiple Traveling Salesman Problem (mTSP) [10], to improve packet transmission and minimize the delay.

While in our study, the heuristic WSN design method is based on Genetic Algorithms (GA), that it is an optimization tool that mimics natural selection and genetics. GA usually operates by locating a global maximum or minimum in a search area with several local maximums or minimums. In previous studies, nonlinear optimization methods, including the GA have been applied to optimizing application specific network deployment [11], and several hierarchical routing protocols, similar to LEACH [12].

Although there are already some several tools for designing a WSN, most of them do not consider the chosen protocol of communication or just ignore the network organization. So, the existing design tools are not oriented towards providing a complete package of solutions based on current search studies. The potential solutions offered are not based on evaluating the performance of a WSN. So, some of the most studied deterministic and deductive WSN design tools are Tinker, SensDep and ANDES. Tinker [13] is a high-level design tool for sensor networks that uses simulated data streams based on real sensor network models, to make the decision on the data processing algorithms that are going to be used. It does not require (or allow) users to specify details such as routing algorithms or retransmission policies, freeing system designers to rapidly iterate among different broad designs before fleshing out details of the one that looks most promising. SensDep [14] as a software design tool incorporates several solution strategies to optimize sensor networks cost and coverage. It uses a deductive method to generate a list of applicable network models that fit the application environment and the parameters of the available sensor nodes and gateway. This tool also takes account of the impacts that the environment has on generating network traffic. ANDES [15] is a developed WSN design tool based on the expansion of the AADL/OSATE framework. AADL/OSATE is a framework that enables hardware and software modeling of embedded systems, as well as the interaction between the various components generated. ANDES enables designers to systematically develop a model for the sensor network, recalibrating system parameters to find an optimal solution based on current performance estimation techniques of such a network. The demand for a tool that can help to design the topology of a WSN, selecting the set of protocols before practical implementation, is increased. Limitations on available energy, specific characteristics depending on the field and the purpose of the application make it quite challenging to design a WSN. The optimal design of the WSN before implementation in the environment is critical and often requires compromises between different competing objectives.

Another recently approach is presented at [16], proposing a Dynamic Load Balance Clustering Mechanism (DLBCM) that not only considers the loading of the CM, but also monitors the energy consumption of the CM in each cluster. This way it is avoided the CM re-election and cluster reconfiguration frequently to keep the entire network topology more stable and efficient. Then, the lifetime of WSN will be prolonged. There are considered also four weights to ponder: the residual energy, the processor utilization, communication bandwidth of node and the distance to the center of the cluster. Their values are changed according with the importance of each item in different specific applications, and there can be some cases when these weights can be set to zero.

This paper considers the issues described above and proposes a WSN-based design algorithm based on GA, which can help WSN's designers in configuring parameters to have the required performance before implementation of the system. So, a deductive design tool for the topology of WSN with hierarchical organization, based on GA, is proposed. In our application, GA operates as a WSN topology design tool by autonomously generating a hierarchical cluster-based network organization by determining each node's position in the distribution area, node's operation state and cluster organization of the active nodes. The GA based design tool operation is bounded by the application specific design requirements and network characteristics related to network coverage, connectivity, energy consumption efficiency and network lifetime.

The main goal of the paper is to describe the algorithm used to generate the topology of a WSN according to the design requirements and to evaluate the performance of the network developed in relation to the application requirements. So, between hundreds or thousands of ways of organizing the network, it is intended to develop the topology and determine the role of nodes, in such a way as to optimize the specific design parameters.

The remainder of the paper is organized as follows: Section 2 discusses the related works for most effective GA applications in WSN, including hierarchical routing protocols and some network distributions methods based on GA; Section 3 presents the applied algorithm, while the weight coefficients used to study the optimization of the network can be found in Section 4. The last Section 5 will bring the conclusions of this work.

2. **Related Works.**

Clustering is usually a highly efficient technique [17,18], where sensor nodes are grouped to form a cluster managed by the Cluster Head (CH). The CH gathers the data, compresses it and sends it to the sink. Thus, the nodes reduce their communication compared with the situation when data are forwarded directly to the sink. Although most of the cluster-routing protocols aim to equally balance the load between sensor nodes, through applying a probabilistic model to electing the CH node each round, they fail to guarantee that the elected node is the best available. There is plenty of room for improvement. Although many clustered based protocols are available in the current literature, only a few well-known protocols based on LEACH are discussed here based on the interest of our work.

2.1. **Cluster-based routing protocols.** LEACH [19] is a typical example of the adaptive clustering routing protocols. Like most of the hierarchical protocol, the operation of LEACH consists of two main phases: the *set-up phase* and the *steady data transmission phase*. In the set-up phase, the CH is elected from the available sensor nodes based on a probabilistic model, and several clusters are constructed dynamically. In the steady data transmission phase, sensor nodes in each cluster send their data to the dedicated CH, that compresses the data and sends it to the destination sink node. LEACH protocol periodically elects the CH nodes and re-establishes the clusters according to a round time, which ensures that the energy dissipation of each node in the network is relatively uniform. Although LEACH protocol distributes the load equally on each CH, still there are some pitfalls. Firstly, there is no guarantee that the selected CH is the best available, for instance, if the elected CH is located near the boundary of the network, other nodes could spend more energy to transferring the message to CH. It is also not possible to determinate a fixed number of CHs elected in each round. Over time, several protocols and methods based on LEACH have been proposed, usually with a considerable improvement for network lifetime [20]. Cluster-based routing protocols generally focus on optimizing the energy efficiency of the network operation, but since WSNs are used for a specific application, there may be several QoS requirements.

2.2. **Genetic algorithm based routing protocols.** Previous studies have demonstrated the usefulness of GA based methods to optimize routing performance in WSNs [21]. Most of the GA applications in WSNs focus on lifetime and energy consumption optimization. The improvement is achieved through the implementation of a GA based algorithm in almost every operational stage of WSNs including node distribution, network coverage, clustering, and data aggregation in order to generate a satisfying set of performance parameters for different WSNs organizations. LEACH-GA [22] is one of the first proposed GA-based adaptive clustering protocols. The objective of this protocol is to optimize the CH selection probability model in order to achieve considerable performance improvement in terms of network lifetime. The proposed GA-based protocol is based on LEACH and operates almost like the standard LEACH protocol. Basically, it includes a *set-up* and a *steady-state phases* for each round in the protocol, which operate exactly as described at LEACH, but it differs since this protocol incorporates in its operation an additional *preparation phase*. This is performed only once before the set-up phase of the first round. Before the first round as the networks begins to operate, in the preparation phase all nodes initially perform CH selection based on a random model. So, each sensor node generates a random number from the interval $[0, 1]$, and then the generated value along with nodes ID and geographical position is sent to the Base Station (BS). Only the sensor nodes, which value exceeds a certain threshold will be considered as a CH candidate. As the BS received the messages from all nodes, it applies a searching algorithm based on GA in order to determinate an optimal probability of nodes being CH. The selection probability for each node is generated based on minimizing the total energy consumption required to complete one round. In the end of the preparation phase, BS broadcasts an advertisement message with the optimal generated probability values to the all nodes in order to form clusters in the following set-up phase. The processes of following set-up and steady-state phases in every round are the same as LEACH.

In other words, the preparation phase generates the probability values for CH selection of the set-up phase, which leads to minimal energy consumption. The proposed GA-based adaptive clustering protocol effectively produces optimal energy consumption that extends the network lifetime. Other clustering protocols based on LEACH include Genetic Algorithm Based Energy Efficient Clusters (GABEEC) [23], the improved LEACH [24] and C-LEACH [25]. All the proposed methods are focused in optimizing the distribution of energy resources through improving the CH selection method.

3. **The Proposed Algorithm.**

Applying the GA to a problem consists of four main steps. The first step is coding the problem, and during this process the structure of a potential solution in the *genome* is built. Genome is a series of binary or alphanumeric characters that need to be manipulated by the GA to evolve candidate solutions, in the hope that they will be more optimal. The second step is to build the *fitness function*, which has a direct impact on the quality of the solutions generated as well as on the complexity of the GA. In our case the *fitness function* is composed of a set of algorithms, the purpose of which is to estimate the parameters of the candidate's topologies. Third, the function of selecting individuals for recombination should be chosen by probability. Finally, the fourth step is to determine the genetic operations that have an impact on diversity, the quality of the solutions generated, and the convergence of the group of candidate solutions toward the global or local maximum or minimum. Genetic mechanisms include recombination and mutation. GAs are random search-based deductive techniques, which means that the algorithm seeks in a field of potential solutions for a global maximum or minimum. In many cases, for a variety of reasons, the GA may prematurely converge to a local maximum.

This can come from choosing the wrong genetic operations. However, another reason is the way the problem integrates into the *fitness function*, as it is the mechanism that directs the algorithm toward optimal local or global solutions. Therefore, before applying GA to solving a problem, it is initially very important to evaluate and decompose the problem in such a way that it can be integrated into a *fitness function*. For the application of the GA in constructing a topology, the parameters of the network we intend to optimize should be defined, as variables of the fitness function.

3.1. **WSN model.**

During this study a homogenous WSN is considered, where nodes are organized in a hierarchical cluster-based network model. Each node can operate either as CH node or as a sensor node. In a WSN, CHs collect the data from respective sensor nodes and forward the aggregated data to a BS at periodic intervals, known as operation rounds. Since CHs are characterized by a high-energy consumption due to its complex and demanding tasks, a major challenge in WSN topology design is to select appropriate CHs in order to optimize network lifetime. WSN's nodes will be distributed on a two-dimensional square network, $X * X$ units (see Figure 1). The area divided into grids is separated by a predefined Euclidian distance and the nodes are placed at the intersections of grids. CHs' (green circles) sensing range is $\sqrt{2}/2$ units and their transmission range is $2\sqrt{2}$ units. As presented in Figure 1 other nodes are operating in active state as: Low Range Node (LRN-cyan circles) and High Range Node (HRN-blue circles). The LRNs' sensing and transmission range is $\sqrt{2}/2$ units, and the total operation energy per round for this state is the lowest of all possible states. The HRN has the transmission and sensing range, $\sqrt{2}$ units, twice as the LRN, and its total operation energy per round is higher than the previous state. Whereas, Inactive Nodes (IN-nodes, X) do not perform any process at all; as a result the node's energy consumption is none. The separation of active sensor nodes in the LRN and HRN is done for energy optimization, coverage and reducing overlaps. Nodes that are close to a dense area and to CH, tend to pass to low range mode to save energy.

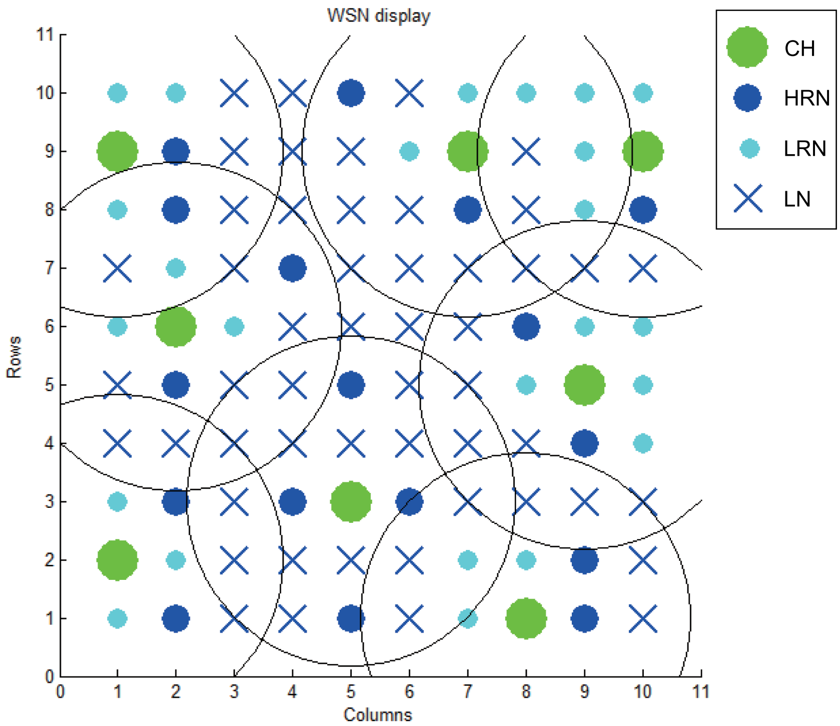

FIGURE 1. Network model layout

On the other hand, nodes located near an uncovered area or away from the CH tend to operate in the HRN state, to provide coverage and to maintain links with the CH. Active nodes outside the coverage area, Out of Range Nodes (ORN), cannot communicate with the CH. They consume energy only for their operation and since the values measured cannot be transmitted to the CH, it would be more efficient to pass them in *sleep* mode.

3.2. **WSN design parameters.** As previously mentioned, the WSN topology design tool's objective is to simultaneously optimize several application specific network performance parameters related to: area coverage, connectivity, energy efficiency and lifetime, through cluster forming and altering the nodes' operation state and position in the monitoring area.

Area coverage is an indicator of the network distribution's effectiveness. In almost every WSN scheme, it is crucial to achieve complete coverage of the region with minimal implementation cost. Hence, the design tool will prioritize the topologies with a lower number of sensor nodes and a higher area coverage. Area coverage is measured and evaluated via the total uncovered surface parameter, U_S. The total uncovered surface is defined as

$$U_S = \frac{\sum U_a}{X \cdot X} \tag{1}$$

where U_a – an area with an inactive node present, but uncovered by any adjacent node, X – height and width parameter of the total network deployment area.

The GA algorithm uses as a deductive technique the search for the most optimal topology for the application in question, through the selection and continuous combination of the most performing topologies between randomly generated groups. The quality assessment of generated or selected topologies is performed through the *fitness function*, which consists of a set of functions that assess network parameters such as coverage, number of sensor nodes per CH, total average energy of the system, minimum and maximum energy for nodes, residual energy, number of nodes out of coverage, number of overlaps and total network lifetime.

For each individual, i.e., topology, of the population of a certain generation, the *fitness function* is applied, which is assigned by a fitness value depending on the result of the respective parameters and weights. In this case, the GA serves as the function of minimizing the function of fitness, which means that topologies with lower fitness value will be favored by the selection function to be selected in order to recombine them to obtain young individuals of the following population.

The application of the global optimization toolbox through Matlab's genetic algorithms consists of three key steps: building the sequence of genome characters based on the problem, determining genetic operations, and constructing the fitness function. The process of coding potential problem solutions in the genome has been addressed above and binary coding has been defined to represent all network nodes. In order to activate the GA toolbox, the syntax code line applied:

$$[P, FitVal] = ga(@FitnessFunction, Individual\_Size, options) \tag{2}$$

where "*FitnessFunction*" is the fitness function, which performs the assessment of population individuals for each generation. "*Individual_Size*" determines the genome length of an individual, which in our case is

$$G_L = 2N^2 \tag{3}$$

The algorithm operates through genetic operations favoring the choice for recombination of individuals with the lowest fitness value. In this way, through the minimization of the fitness function, topologies with optimized performance parameters for application requirements can be obtained.

Parameters of the "options" structure for configuring the GA will be presented in the following in the modality:

- *Option* (value) – Description.
- *PopulationType* (bitstring) – Determining the type of genome on which the GA will be applied, the chosen model is the binary row.
- *Generations* (4000) – Determine the maximum number of iterations or generations of the GA before its termination. The applied algorithm will be terminated after 4000.
- *FitnessScalingFcn* (fitscalingprop) – The method of scaling individuals of the population based on fitness values is determined. In this case, the selectivity of the selection is in direct proportion to the value of the fitness.
- *SelectionFcn* (selectionstochunif) – Determine the selection function of individuals for recombination. In the applied case, the method of universal stochastic selection was chosen.
- *CrossoverFcn* (crossoverscattered) – The genetic recombination operation of selected individuals for the generation of individuals of the following population is selected.
- *MutationFcn* (mutationgaussian) – The mutation technique used is determined.
- *StallGenLimit* (4000) – Terminate the GA if for a certain number of generations, 4000 in this case, there is no progress in the average fitness value of the population.
- *StallTimeLimit* (10000) – Terminate the GA if over a certain period, measured in seconds, 10000 in this case, we have no progress in the average fitness value of the population.

The pseudo code of the *fitness function* will be described in the following.

Pseudo code of fitness function

Step 1: Decode the genome of the individual m from the population $M(t)$ and construct the matrix of the structure with the data of the positions and states of the nodes in the network;

Step 2: Build the connection matrix depending on the distances of the CH from the sensor nodes, based on the structure matrix;

Step 3: Based on the link matrix evaluate:
- a) Sensor nodes density for CH, S_pC;
- b) Number of non-connected nodes, SOR;
- c) Overlap number, O_v;
- d) Communication energy for each node;

Step 4: Evaluate the uncovered U_S surface, testing whether the area of inactive and unconnected nodes is covered by adjacent connected nodes;

Step 5: Build the power matrix for each active node, through the amount of operating and communication energy;

Step 6: Based on the energy matrix estimate:
- a) Average energy, E_{mA};
- b) Minimum energy consumed per node, $E_{\min}$;
- c) Maximum energy consumed per node, $E_{\max}$;
- d) Full load network life, L_T;

Step 7: Calculate the fitness value F of the individual m through the weighted sum function:

$$F = A_1U_S + (-A_2)S_pC + A_3E_{mA} + A_4SOR + A_5O_v + (-A_6)E_{\min} + A_7E_{\max} + (-A_8)L_T;$$

Step 8: Repeat the above maps for all individuals' m of the population.

The *fitness function* is not a ready-made feature implemented in Matlab's optimization toolbox. The determination and initialization of the variables of the *fitness function* and the methods of their evaluation for each individual, are performed through the functions implemented by the user, as they are specific depending on the problem that is being treated. In the design application of a WSN topology, the GA of the global optimization toolbox acts as a minimization function for the *fitness function*. As shown above, the GA, after generating a population, calculates the value of fitness for each individual of the population through the function of the weighted sum of the network parameters, Step 7. Subsequently, the algorithm operates through genetic operations favoring the choice for recombination of individuals with the lowest fitness value. In this way, through the minimization of the *fitness function*, topologies with optimized performance parameters for application requirements can be obtained.

4. **Simulations.** In the preceding section, it is described the methodology used and the implementation of the deductive design tool for WSN topology based on GA. The present section describes first the performance and application requirements for generating a set of optimal topologies to meet the requirements of a standard WSN. The results will then be interpreted and evaluated in order to select the most appropriate network model for the case. Finally, based on the results obtained, the performance evaluation of the WSN topology design tool will be evaluated, based on the optimization of the performance parameters.

4.1. **Performance and application requirements.** As has been mentioned before in this paper, the design of a WSN topology has to take account of the specific performance requirements of the WSN, which vary depending on the application. The priority before implementation of the current network in the environment is a compromise between different parameters of WSN, such as connectivity, coverage and energy efficiency. The algorithm performs this function independently, by determining the specific position of the nodes in the network, their states or potential roles, and organizing the nodes into clusters. Before applying the design tool, we first need to assess the capabilities and requirements for the target network and prioritize each of the performance parameters. In the network design tool, the user has the option of assigning importance to each parameter depending on weights it assigns to those, at *fitness function* of GA. Also, for the application in question, the user must specify the network characteristics such as node power capacity, communication radius, coverage radius, operational and communication energy costs per round, communication radius and the unit of surface on which the network will monitor. Once the performance parameter weights and network characteristics are set, then the application of the network design algorithm can be started. The assignment of algorithm execution conditions, including genetic operations, their configuration and termination criteria are experimentally derived during the construction of the algorithm. During this study it is chosen a homogeneous topology for the hierarchical organization that is applied in most applications. Usually, WSNs have three main priorities: full environmental coverage, ensuring sufficient connectivity to carry out communication functions, and optimal lifetime for the assigned task. The network characteristics for the application in question are summarized in Table 1.

Table 1. Network design criteria

WSN design criteria	**Value**
Surface for coverage	10×10 unit of surface
Maximum number of nodes	100
Energy capacity	1000 unit of energy
Operational energy for LRN	4 unit of energy
Operational energy for HRN	8 unit of energy
Operational energy for CH	16 unit of energy
Transmission radius for LRN	$\sqrt{2}/2$ unit of length
Transmission radius for HRN	$\sqrt{2}$ unit of length
Transmission radius for CH	$2\sqrt{2}$ unit of length
Coverage radius for LRN	$\sqrt{2}/2$ unit of length
Coverage radius for HRN	$\sqrt{2}$ unit of length
Coverage radius for CH	$\sqrt{2}/2$ unit of length
Communication energy per round	$0.6 * d^2$ (d – distance between two nodes)

Having already settled the capacities of the nodes and the characteristics of the network intended to be designed, priorities must be set. So, the weight coefficients of each parameter at *fitness function* have to be decided. GA is a random-based classical evolutionary algorithm. Therefore, its application to the simultaneous optimization of a set of interrelated performance parameters generates a wide range of potential solutions. Selecting the solution set that fits the application is a relatively complicated process, with no predetermined selection method. For each application there may be a set of weight combinations, which can generate results that meet or not most of the criteria set. The goal at this stage is to select weights of parameters by testing different combinations of variables coefficients on the fitness function for identifying the highest performing combination for the selected application. To determine the combination of weights that provide the best set of solutions, we will rely on eight parameters. However, among them, five are the most important, that will serve as criteria for evaluating the performance of generated network topologies.

The first criterion selected is the uncovered surface (U_S) of the simulated environment, which is measured related to the total area of the environment. Combinations of weights which reduce the uncovered territory will be considered more favorable. The second criterion is the residual energy (E_R) in the active nodes after the network's failure, which reflects on the efficiency of using the energy capacities of the active nodes. A small value of this parameter indicates that we have efficiency in using network resources. The next criterion is lifetime (L_T), the most important parameters for evaluating the network's energy efficiency. The network must operate in the testing environment for a specific period in order to successfully implement the defined monitoring functions. The last two criteria are related to connectivity: the Node Degree (ND) after the network is disconnected and the number of overlaps (O_v). ND helps to estimate how much is affected the network's connectivity after the death of a CH in the network and how much the network's recovery is possible, while O_v determines the efficiency of nodes distribution across clusters.

4.2. **The results of the network design algorithm.** Designing the optimal topology of a WSN involves a set of performance parameters related to each other and the main objective lies in setting a compromise between these parameters to meet application requirements. Deterministic techniques are not successful in these applications; therefore, the use of GA was proposed and applied, as a deductive method for constructing the topology with the best performance. Network's performance parameters are set, based on the WSNs' model with hierarchical organization and homogeneous nodes. Performance parameters, grouped by impact on coverage, energy efficiency, and connectivity parameters, are integrated as highly weighted by *fitness function's* variables. Each performance parameter can be assigned a specific weight based on the importance it has for a specific application. Generation of performing solutions is realized by applying the GA to minimizing the *fitness function*.

The weights of each parameter in the WSN design algorithm are arbitrarily assigned in the beginning according to the application requirements. Their assignment is accomplished through the method of testing and excluding inefficient cases for the application. Table 2 presents the performance parameters symbols and their perspective values to be obtained from the GA. At the end of simulating different topologies with optimal performance, for each combination case, 100 individual tests were performed and the averaged results are summarized in Table 3. The first test is the situation where all parameters have the same unit weight. Subsequently, based on the results of the specified parameters, appropriate measures were taken, increasing or decreasing the weight value. It is done the same in the following cases, until the combination that generates the most optimal solution is determined. All the weight coefficients are units at the beginning of simulations except the maximum and minimum energy, whose impact is neglected. Since the priority in our application is coverage, in this case the value 0.3 of U_S, does not meet our criteria. Consequently, in the second case, U_S value is 2, by increasing its impact on fitness. Now there is improved coverage and slight changes of other parameters, but O_v is still high and is not optimal for our case, which indicates that there is inefficient distribution of sensor nodes across clusters. So, O_v is set to 2, to limit the value of overlaps. Based on

Table 2. Performance parameters with their respective weight coefficients

Performance parameter	Weight coefficient	Objective of GA
Uncovered surface (U_S)	A_1	Min.
Sensor nodes density for CH (S_pC)	A_2	Max.
Average energy (E_{mA})	A_3	Min.
Number of non-connected nodes (SOR)	A_4	Min.
Number of overlaps (O_v)	A_5	Min.
Maximum energy ($E_{\max}$)	A_6	Max.
Minimal energy ($E_{\min}$)	A_7	Min.
Lifetime (L_T)	A_8	Max.

Table 3. Results of combinations of weights coefficients

No.	Weight coefficient								Parameters of performance				
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	U_S	L_T	E_R	ND	O_v
1	1	1	1	1	1	0	0	1	0.3	43	614	38	0.4
2	2	1	1	1	1	0	0	1	0	42	621	39	0.6
3	2	1	1	1	2	0	0	1	0.1	40	631	38	0.1
4	2	2	1	1	2	0	0	1	0.2	37	662	41	0.5
5	2	0.5	1	1	2	0	0	1	0.3	41	618	33	0.2
6	2	0.5	1	1	2	0	0	2	0.5	45	577	33	0.6
7	2	0.1	1	1	2	0	0	2	1	48	446	19	0.7
8	2	1	1	1	3	0	0	1	0.2	39	639	37	0
9	2	1	2	1	2	0	0	1	0.2	40	621	34	0.1
10	2	0.5	2	1	2	0	0	1	0.5	43	494	18	0.3
11	2	1.5	1	1	0.5	0	0	1	0	42	631	43	1.1
12	2	0.5	1	1	1.5	0	0	1	0.1	42	606	34	0.5
13	2	1.5	0.5	1	1	0	0	1	0	40	641	41	0.4
14	2	0.5	1.5	1	1	0	0	1	0	46	478	21	1.1
15	2	1	0.5	1	1.5	0	0	1	0.1	40	635	39	0.2
16	2	1	1.5	1	0.5	0	0	1	0	43	606	38	1.8
17	2	1	1	1	2	1	0	1	0.4	44	595	36	0.4
18	2	1	1	1	2	0	1	1	0.2	40	636	38	0.1
19	**2**	**1**	**1**	**1**	**2**	**0.5**	**0**	**1**	**0.2**	**42**	**612**	**36**	**0.1**
20	2	1	1	1	2	0	0.5	1	0.1	40	632	37	0.2

the results of the following case, the coverage criterion is slight changed and in this case the overlaps criterion is met.

A method to improve the distribution of sensor nodes is to change the weight value of S_pC or the maximum power parameter. The algorithm will favor the generation of topologies where there is a uniform distribution and with less CH. Thus, this way is applied the method for the selection of the best combination of coefficients for the network considered.

The simulation shows that to obtain a full-coverage WSN, the combination of weights that offers the most optimal parameter's compromise is the case 19. As can be seen, unlike other cases, it is set a relatively low value of A_6; as a consequence this limits the number of sensor node connections per CH. This combination of weights enables to increase L_T, and it means more efficient use of energy resources. The distribution of the sensor nodes in clusters is very efficient for full-coverage of the environment, for the uniform power consumption between CHs and to avoid overlaps. All the parameters and characteristics of optimal case 19, as an average test of 100 cases, are summarized in Table 4.

Table 4. The average parameters of the most optimal topology

Network parameter	Values
Number of CH	7
Number of HRN	18
Number of LRN	24
Number of IN	51
Uncovered surface (U_S)	0.2
Sensor nodes density for CH (S_pC)	6
Number of overlaps (O_v)	0.19
Maximum energy ($E_{\max}$)	24
Minimal energy ($E_{\min}$)	1
Lifetime (L_T)	42
Average energy remaining after disconnection of the network	611.3
ND before network disconnection	42
ND after network disconnection	36

Based on the generated topology results, it is visible that for full coverage of the environment only 49 active sensors out of 100 are needed. The nodes are scattered across the environment in 7 clusters with 6 sensor nodes per each CH. The distribution of sensor nodes per CH is quite uniform and efficient, as there is no overlap (0.19). Also, uniform distribution of sensor nodes enables balanced power consumption between CH and nodes with higher power consumption. Compared to other tested situations, the network with the generated topology has optimal L_T, as it can operate for 42 rounds, before the network is disconnected. After the network failure, 40% of the energy is consumed and 36 out of 42 connections between the nodes are available, so the connectivity is high enough to enable any recover technique.

GA are deductive techniques searching for a global maximum or minimum in a space with several choices. So, the best possible solution in our case would be at the global minimum. To assess whether we have really generated the topology with the highest performance for the required situation, we need to evaluate the performance of the algorithm. The performance of a deductive algorithm can be assessed based on the progress of the fitness value of the individuals generated during its implementation. This assessment method may again be insufficient to avoid or detect premature convergence and therefore monitoring of specific application parameters during algorithm implementation is needed, if the parameter values converge or not, to acceptable values for the application. For the topology of a WSN it is sufficient to estimate the progress or regression of performance parameters to determine whether the applied algorithm is effective in generating a potential solution or not. In case the algorithm converges prematurely, in most cases we are dealing with a narrow search space. Simple measures that can be taken to avoid it are increasing the rate or changing the method of mutation, changing the method of recombination, selecting individuals for recombination, increasing the number of individuals per population, and implementing the algorithm over more generations.

The selection and configuration of genetic operations are performed experimentally through evidence based on the progress of performance parameters. Figure 2 presentes the minimization of the number of overlaps (O_v), like predicted in Table 2, throughout

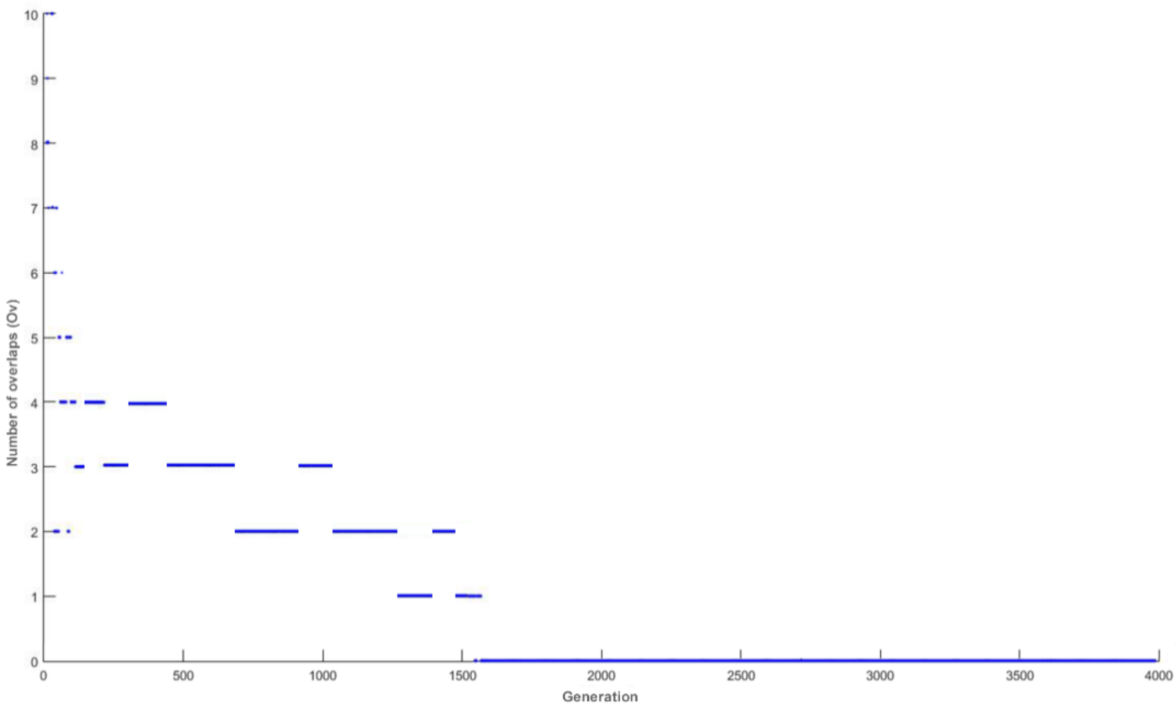

Figure 2. Number of overlaps (O_v)

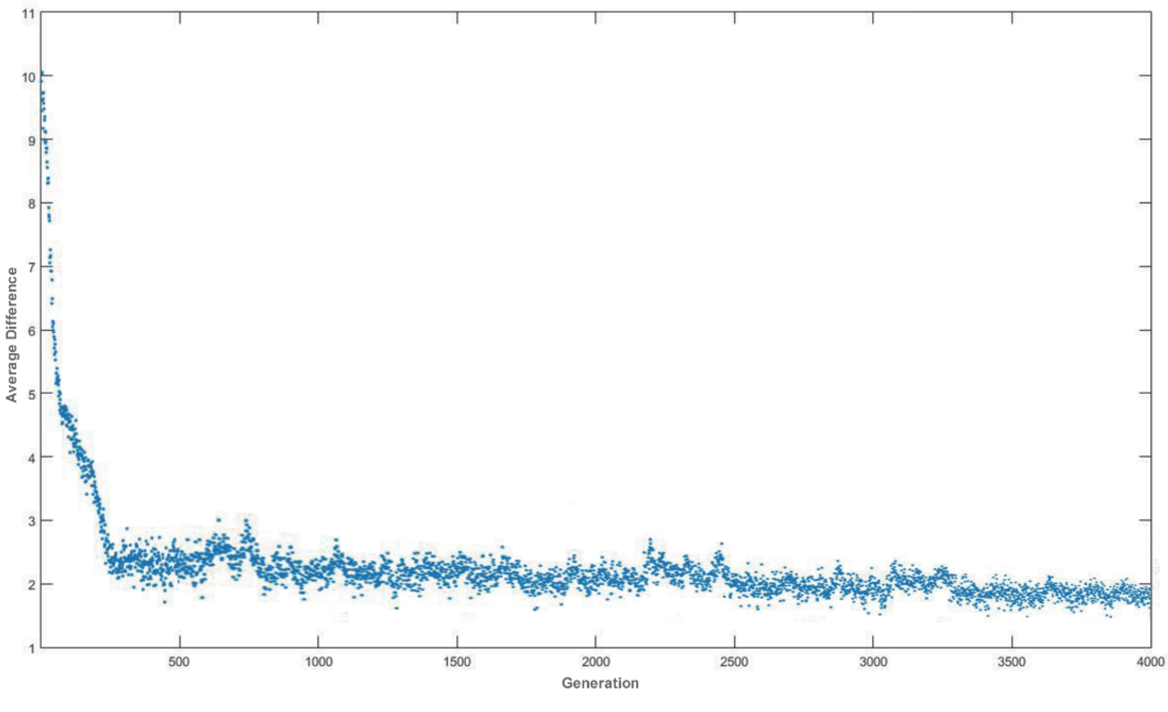

Figure 3. Average difference of the value of *fitness function* between nodes during the application of GA

the implementation of the algorithm with the combination number 19 of the weights (Table 3). The results of Figure 2 confirm once more the results shown in Table 4, where in average the number of overlaps (O_v) is 0.19. Based on the results of the GA applied to the design of a WSN, as can be seen in Figure 3, we have a convergence of population individuals difference for the *fitness function*. At the beginning of the algorithm, when we have the first generation is generated randomly, the value of this difference is very high, which means that randomly generated solutions are far from the optimal required solution, and the average difference in the value of *fitness function* among individuals of the population is quite high. Very soon after several dozen generations, the algorithm begins to converge towards more acceptable solutions, with less value of the *fitness function*. However, it can be noticed that even after convergence, the difference between the individuals of the population is high, so the search space is wide enough to enable the generation and selection of the best solutions through genetic operations. Over the generations, the algorithm has made progress until it reaches a limit at which the difference between individuals in the fitness value decreases and we have no progress in the *fitness function*'s value of the best individual. At this point we can say with certainty that the population has converged and we are very close or we have located the most optimal possible solution. Observing the degree of difference between individuals and the progress of the average fitness value is a fairly good method of determining whether or not we have found the best possible solution. However, often times, this method is insufficient, as we have no data regarding the progress of other parameters. As far as we know, the progress of the fitness value can also come from the improvement of a single parameter of the *fitness function* with high weight ratio in relation to other parameters, while other parameters may not change or may deteriorate. Since the characteristics of the network are directly dependent on the application, the design of WSNs is a process that requires consideration of the application requirements but also of the wireless sensor network limitations. The way joints are distributed in the environment, and the state of their functioning and organization in clusters has a significant impact on the efficiency of communication functions, environmental monitoring and energy use. The network topology design is a process with an impact on network performance and must be carried out before implementing it in the environment. Designing WSNs, due to environmental constraints and requirements, often times is a difficult process that consists of setting a compromise between competing performance parameters.

After the 3000th generation, the progress of the average fitness value and network performance parameters in subsequent populations stopped and remained constant, with few changes in some cases. For this reason, the genetic algorithm is configured to be terminated after the 4000 generation. Based on the results obtained, we can say that the design algorithm of a WSN network is performable in optimizing all network parameters simultaneously and can generate optimal topology depending on the priority assigned to each parameter.

5. **Conclusions.** In the present paper the design of a WSN homogenous network with hierarchical organization is shown, with the priority of covering an environment of minimum cost, high connectivity and maximum lifetime. Designing a network requires setting the optimal compromise between performance parameters, where the priority of each parameter was controlled through the values of the weight coefficients in the fitness function. We have demonstrated that our algorithm is able to find the most optimal combination of weights through continuous testing and case selection, which generates a topology with network parameter values that meet the application criteria. Finally, the chosen combination of weights can be applied to generating the most performable topology possible.

As future works it can be simulated the optimization of communication between nodes. The design criterion may include the selection and evaluation of the efficiency of routing algorithms. This can be accomplished through the implementation of a network performance simulation and evaluation function for certain hierarchical routing protocols.

Another aspect that can be implemented is the simulation of a network failure and the testing of the performance of various network recovery techniques, in order to select the most optimal technique.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine*, 2002.

[2] L. C. Tagliabue, F. R. Cecconi, N. Moretti, S. Rinaldi, P. Bellagente and A. L. C. Ciribini, Security assessment of urban areas through a GIS-based analysis of lighting data generated by IoT sensors, *Appl. Sci.*, vol.10, no.6, DOI: 10.3390/app10062174, 2020.

[3] A. Pedersen and T. F. Brustad, A study on hybrid sensor technology in winter road assessment, *Safety*, vol.6, no.1, DOI: 10.3390/safety6010017, 2020.

[4] M. Al-Zubaidie, Z. Zhang and J. Zhang, REISCH: Incorporating lightweight and reliable algorithms into healthcare applications of WSNs, *Appl. Sci.*, vol.10, no.6, DOI: 10.3390/app10062007, 2020.

[5] A. Kristoffersson and M. Lindén, A systematic review on the use of wearable body sensors for health monitoring: A qualitative synthesis, *Sensors*, vol.20, no.5, DOI: 10.3390/s20051502, 2020.

[6] D. M. Doolin and N. Sitar, Wireless sensors for wildfire monitoring, *Proc. of SPIE on Smart Structures & Materials*, pp.477-484, 2005.

[7] L. Tang, Z. Lu and B. Fan, Energy effcient and reliable routing algorithm for wireless sensors networks, *Appl. Sci.*, vol.10, no.5, DOI: 10.3390/app10051885, 2020

[8] Z. Nazari, M. Nazari and D. Kang, A bottom-up hierarchical clustering algorithm with intersection points, *International Journal of Innovative Computing, Information and Control*, vol.15, no.1, pp.291-304, 2019.

[9] M. A. Sayeed and R. Shree, Optimizing unmanned aerial vehicle assisted data collection in cluster based wireless sensor network, *ICIC Express Letters*, vol.13, no.5, pp.367-374, 2019.

[10] J. Ma, S. Shi, X. Gu and F. Wang, Heuristic mobile data gathering for wireless sensor networks via trajectory control, *International Journal of Distributed Sensor Networks*, vol.16, no.5, DOI: 10.1177/1550147720907052, 2020.

[11] L. J. G. Villalba, A. L. S. Orozco, A. T. Cabrera and C. J. B. Abbas, Routing protocols in wireless sensor networks, *Sensors*, vol.9, no.11, pp.8399-8421, 2009.

[12] J. Díez-González, R. Álvarez, N. Prieto-Fernández and H. Perez, Local wireless sensor networks positioning reliability under sensor failure, *Sensors*, vol.20, no.5, DOI: 10.3390/s20051426, 2020.

[13] J. Elson and A. Parker, Tinker: A tool for designing data-centric sensor networks, *IPSN'06*, TN, USA, 2006.

[14] R. Ramadan, K. F. Abdelghany and H. El-Rewin, SensDep: A design tool for the deployment of heterogeneous sensing devices, *The 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS)*, 2006.

[15] V. Prasad, T. Yan, P. Jayachandran, Z. Li, S. H. Son, J. A. Stankovic, J. Hansson and T. Abdelzaher, ANDES: An ANalysis-based DEsign tool for wireless sensor networks, *The 28th IEEE International Real-Time Systems Symposium (RTSS 2007)*, Tucson, AZ, USA, pp.203-213, 2007.

[16] S.-C. Wang, Y.-L. Lin, M.-L. Chiang and H.-H. Pan, Improve the stability of the Internet of things using dynamic load balancing clustering, *International Journal of Innovative Computing, Information and Control*, vol.16, no.1, pp.63-76, 2020.

[17] X. Liu, A survey on clustering routing protocols in wireless sensor networks, *Sensors*, vol.12, no.8, pp.11113-11153, 2012.

[18] D. Bhattacharyya, T. Kim and S. Pal, A comparative study of wireless sensor networks and their routing protocols, *Sensors*, 2010.

[19] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, *Proc. of the 33rd Annual Hawaii International Conference on System Sciences (HICSS)*, Big Island, HI, USA, pp.3005-3014, 2000.

[20] E. Kotobelli, E. Zanaj, M. Alinci, E. Bumçi and M. Banushi, A modified clustering algorithm in WSN, *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol.6, 2015.

[21] B. Zanaj and E. Zanaj, Review of traveling salesman problem for the genetic algorithms, *Journal of Information Sciences and Computing Technologies (JISCT)*, vol.5, 2016.

[22] J.-L. Liu and C. V. Ravishankar, LEACH-GA: Genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks, *International Journal of Machine Learning and Computing (IJMLC)*, vol.1, no.1, pp.79-85, 2011.

[23] S. Bayraklı and S. Z. Erdogan, Genetic algorithm based energy efficient clusters (GABEEC) in wireless sensor networks, *Procedia Computer Science*, vol.10, pp.247-254, 2012.
[24] P. Nayak and B. Vathasavai, Genetic algorithm based clustering approach for wireless sensor network to optimize routing techniques, *The 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence*, 2017.
[25] A. Rahmanian, H. Omranpour, M. Akbari and K. Raahemifar, A novel genetic algorithm in LEACH-C routing protocol for sensor networks, *Proc. of the 24th Canadian Conference on Electrical and Computer Engineering*, Niagara Falls, Ontario, Canada, 2011.