A Cooperative Coevolutionary Algorithm for the Design of Wireless Sensor Networks

Track name: Bio-Inspired Solutions for Wireless Sensor Networks

André Siqueira Ruela
Universidade Federal de Ouro Preto, Ouro Preto, Brazil
andre.siqueira.ruela@gmail.com

André L. L. Aquino
Universidade Federal de Alagoas, Maceió, Brazil
alla@ic.ufal.br

Frederico G. Guimarães
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
fredericoguimaraes@ufmg.br

ABSTRACT

This work proposes a cooperative coevolutionary algorithm for the design of a wireless sensor network considering complex network metrics. It is proposed an heuristic based on cooperative coevolution to find a network configuration such that its communication structure presents a small value for the average shortest path length and a high cluster coefficient. This configuration considers a cluster based network, where the cluster heads have two communication radii. The mathematical model of the cluster head location problem was developed to determine the nodes which will be configured as cluster heads. This model was adopted within the coevolutionary algorithm. We describe how the problem can be partitioned and how the fitness computation can be divided such that the cooperative coevolution model is feasible. The results reveal that our methodology allows the configuration of networks with more than a hundred nodes with two specifics complex network measurements allowing the reduction of energy consumption and the data transmission delay.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology, wireless communication*; G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, integer programming*; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

Complex Networks, Coevolutionary Genetic Algorithms, Network Design, Wireless Sensor Networks

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [2] represent an emerging technology that allows the monitoring and control of physical and environmental variables and conditions, such as temperature, sound, light, vibration, pressure, movement, and pollution. A WSN consists of a great number of wireless autonomous devices, spatially distributed, called sensor nodes, which work in a cooperative way to perform many different functions. The sensor is usually equipped with a small-sized micro-controller, a radio transceiver or other wireless communication device and an energy source. In WSN applications, sensor nodes are usually not deployed to pre-determined locations. Random deployment of sensors is preferred instead.

WSNs differ in many aspects from conventional computer networks, mainly because they have several resource restrictions, such as low computational power, reduced bandwidth, and limited energy resource. Due to these characteristics, it is necessary to design specific models, topologies and algorithms to circumvent the difficulties related to this technology [12, 22]. Among these resource restrictions, energy consumption is critical. The operation of wireless data communication is the main source of energy consumption: sending one bit demands on average thousand times more energy than other internal operations and data sensing. Therefore, algorithms for WSNs need to be carefully designed. Sending a large amount of data can be problematic, causing excessive delay in response time, thus invalidating the data. Moreover, a large traffic on the network can diminish its lifetime. Due to these restrictions, in some cases, it is necessary to adopt specific infrastructure designs to balance the network requirements while keeping its functionality.

The information about the phenomenon monitored is reported through the network to the sink node [18]. WSNs applications, generally, have n-tier ($n \geq 1$) architecture designs, where the most used is the two-tier [4, 20]. A two-tier architecture consists of several clusters and one or more sinks. Each cluster comprises a number of member nodes, responsible for the sensing task over the corresponding area, and a cluster head is designated to collecting data from the local sensors and routing them to the sink. Such cluster-based architecture offers some inherent advantages against the flat one in terms of energy consumption: (i) only the cluster head nodes are involved in routing task and the local nodes only transmit the sensed data to a cluster head nearby. Thus the energy consumed in data transmission could be substantially reduced; and (ii) considering that only the cluster head transmits data out of the cluster, this helps to save energy by avoiding collision between local nodes [1, 9, 10].

In this paper, we propose a coevolutionary algorithm [13, 15] to design a two-tier WSN considering complex network measurements [6]. The main contributions of this work are:

- The main problem is partitioned into smaller ones, enabling the efficient use of a cooperative coevolutionary algorithm;
- Based on the characteristics of the problem, a new initial population generator was implemented. This generator is based on the centrality betweenness metric;
- The method has a faster convergence, when compared with other approaches. This occurs because of the better set of individuals generated by the population generator.

This work is organized as follows. In Section 2, we present the network design and problem definition. In Section 2.1 we discuss about the problem formulation. Next, in Section 3, we show the cooperative coevolutionary algorithm proposed. Simulation results are shown in Section 4, and Section 5 concludes this study and presents the future work.

2. PROBLEM DEFINITION

As mentioned before, data transmission is the most expensive operation in the network. Therefore, considering the whole network, data propagation (routing) from all nodes to the sink is directly related to the life time of the network. A simple naive, but inefficient, way of propagating information through the network is flooding. In this case, the information is flooded to all sensors until it reaches the sink node [14]. This strategy causes unnecessary communication, consequently, a large energy consumption and a high response time to deliver the data.

A common alternative to flooding is tree routing, a simple and low-overhead routing protocol. Using a tree routing, each sensor is configured to send its data only to a specific sensor node, denoted father node. The choice of which node will be the father depends on the policy established by the application, in general, the shortest path policy is used [16]. The major drawback of tree routing is the increased hop counts as compared with more sophisticated path search protocols. However, there is a significant energy consumption because the link is kept, i.e., all non father nodes perceive the propagated data.

Additionally, considering applications that use thousands of nodes working independently and together [7], some strategies based on tree routing might not be scalable. An alternative routing strategy based on complex network measurements (Figure 1), consists in setting some sensor nodes as cluster heads based on the energy or communication flow, these ones using a communication radius greater than that used by normal nodes. Normal nodes propagate their data to a given cluster head using a normal link frequency, and the cluster heads propagate their data to the sink node using a special link frequency. In both cases it is used a multi-hop communication. The use of these cluster heads leads to important characteristics of complex networks: a small average shortest path length between all sensors and the sink; and a high cluster coefficient, see [21]. This complex network characteristics help us saving network resources, avoiding excessive communication, and reducing the time to data delivery.

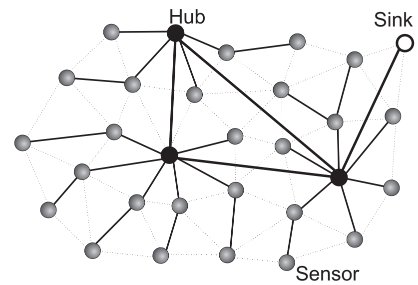

Figure 1: Complex network propagation.

In fact, the proposed approach gives rise to a complex network, which is a network with irregular, complex and dynamic structure [5]. The theory of complex networks provides a mathematical framework for analyzing a number of real-world networks that otherwise could not be addressed with the available traditional models. The theory of complex networks can be useful in the study of WSNs, given some of its peculiar characteristics, such as the quantity, increase, and distribution of nodes in the network.

Based on these aspects, the problem addressed in this work can be stated as follows: *Consider a geometric graph, $G = (V, E)$, where V represents the set of sensor nodes and E the set of edges, representing the logical links between nodes. These links are determined considering the communication geometry of the node, i.e., all neighbors reachable by the node $v \in V$. The problem is to find the better nodes $v \in V$ that should enable the auxiliary radii, and hence be configured as cluster heads, generating a new set of edges E' and therefore a new network $G' = (V, E \cup E')$, such that G' can be characterized as a complex network with a small average shortest path and a high cluster coefficient.*

Therefore, the main hypothesis considered over the problem is: *Characterizing the WSN physical layer as a complex network allows us to build a two-tier WSN logical topology that minimizes the energy consumption and delay.*

In this paper only the complex network measurements will be verified and evaluated in the physical layer. The verification of the logical topology performance will be reported in future work.

2.1 Problem Formulation

In this section, the conventional cluster head allocation problem formulation is described. Initially, consider the mathematical formulation of the single allocation problem in sensor networks, which is an approach to the problem stated above.

Given a network with a set of nodes, the problem consists of finding the nodes that will be reconfigured as cluster head and also the logical links that should be established in order to minimize the total cost. Let N be the set of normal nodes and H be the set of cluster head, such that $N \cup H = V$ and $N \cap H = \emptyset$.

The parameters of our mathematical model are: ϕ_i is the communication demand, i.e., the total amount of data that node i must send to the sink; r is the basic communication radius; d_{ij} is the distance between node i and node j; c_{ij} is the fixed communication cost per data unit from node i to node j; a_j is the fixed installation cost of node j as a cluster head. It is inversely proportional to the distance between j and the sink, i.e., the higher the distance from j to the sink, the lower the installation cost.

The decision variables of the mathematical model are $z_i \in \{0,1\}$, $z_i = 1$ if node i is defined as cluster head, and $z_i = 0$, otherwise; and $q_{ij} \in \{0,1\}$, $q_{ij} = 1$ if there is a logical link between nodes i and j, and $q_{ij} = 0$, otherwise.

A nonlinear integer programming formulation of the problem defined above is given by:

$$z^* = \arg\min \sum_{i \in V} a_i z_i + \sum_{i \in H} \left(\sum_{j \in N} \phi_j c_{ji} q_{ji} \right) + \\ + \left(\sum_{k \in H} (c_{ik} q_{ik} + c_{k0}) \right) \times \sum_{i \in H} \left(\sum_{j \in N} \phi_j q_{ji} + \phi_i \right) \qquad (1)$$

The objective function (1) gives the total cost for establishing the two-tier network. This total cost includes the installation cost, first term in (1), and the propagation cost. The second term in (1) expresses the propagation cost of data from all sensors to their corresponding cluster heads, while the third term represents the total amount of data that the cluster head $i \in H$ must send to the sink, which is multiplied by the cost of sending the data to the sink through cluster head $k \in H$, where k may be equal to i or may be a different cluster head. This results in two or three hops from any node in the sensor network to the sink.

The objective function is subject to the following constraints:

$$\sum_{j \in V} q_{ij} = 1, \qquad \forall i \in N \qquad (2)$$

$$\sum_{j \in H} q_{ij} \leq 1, \qquad \forall i \in H \qquad (3)$$

$$q_{ij} \leq z_j, \qquad \forall i \in N, \forall j \in H \qquad (4)$$

$$d_{ij} q_{ij} \leq 2r, \qquad \forall i \in N, \forall j \in H \qquad (5)$$

$$d_{ij} \leq 3r, \qquad \forall i \in H, \forall j \in H \qquad (6)$$

$$z, q \in \{0,1\} \qquad (7)$$

which are detailed as follows: the constraint (2) guarantees that a node $i \in N$ is connected to only one cluster head; the constraint (3) guarantees that data from the cluster head i is either routed through one cluster head or directly to the sink; (4) ensures that data from node $i \in N$ is only routed through a cluster head node; (5) Ensure that the distance between the node i and a cluster head j is less than or equal to twice the communication radius; (6) Ensure that all cluster heads are within the range of all other ones, using three times the value of the communication radius; and finally, (7) restrict the values of the integer variables z_i and q_{ij} to assume either 0 or 1.

It is important to highlight that, in our case, it is expected that the data propagated through to sink pass only through two cluster heads. In additional, there is none cluster head election strategy that consider the energy and flow parameters to optimize the global topology.

3. COEVOLUTIONARY ALGORITHM

Coevolution can be classified into two types: competitive and cooperative. Basically, in competitive coevolution, there is something similar to an "arms race" in which one population competes with the other [3], [17]. In this model, the fitness of an individual is directly related to its ability to stand out in competition with individuals of other species that evolve in parallel in their populations. In cooperative coevolution, a complex problem is divided into smaller and simpler modules that evolve individuals separately. These individuals are combined to form a single solution suitable for the original problem. The fitness of an individual depends on its ability to cooperate with individuals from other sub-populations. Obviously, the more an individual is able to cooperate to develop a good overall solution, the greater its probability of selection among the remaining individuals. This strategy has shown good results in solving complex problems [13], [15].

Given these characteristics, we develop a heuristic-based Cooperative Coevolutionary Algorithm (CCA) to solve the problem presented in the previous section, which is the design of the physical structure in a WSN based on complex network theory, that would improve the performance of the tree based logical topology to be created over this physical topology.

Two important steps in the definition of any coevolutionary model are (i) an adequate division of the problem and (ii) a suitable encoding of the individuals.

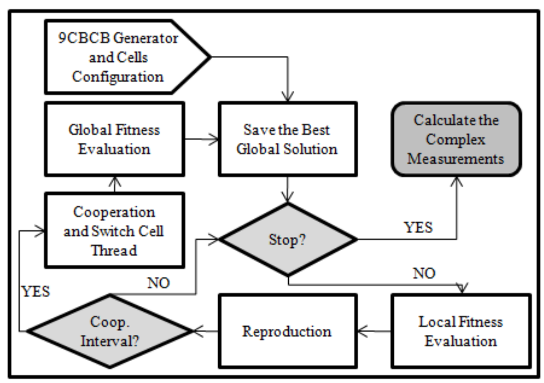

Figure 2: Cooperative Coevolutionary Algorithm Flowchart

Figure 2 illustrates the CCA. The first step of the algorithm is to load the random geometric graph and generate the standard population. Then, the problem is divided into L cells, which each cell run in a thread and has it's own subpopulation. The cells are positioned into an execution queue. The first cell of the queue execute its local evolutionary operators until the cooperation interval. During this interval, the cell update its local data to cooperate with others, producing the global solution. Then, the running cell is positioned at the end of the queue and the next cell in the queue start its execution. This process repeat until an stopping criterion is reached. Finally, the best solution ever found is returned and the complex measurements are calculated over it. Next subsections describe in more detail the division of the problem for cooperation, the basic operators used and the initial population generator developed for the problem.

3.1 Dividing the problem for cooperation

The global objective function (1) was explored by a memetic algorithm in our previous work [19]. Observing the solutions obtained by the memetic algorithm for all the problem instances, we can see that there is not a large number of cluster heads. The algorithm keeps a balance between the high installation cost of nodes as cluster heads and the propagation cost. Figure 3 is an example of a solution returned by the memetic algorithm, illustrating the physical layer evaluated by the global objective function.

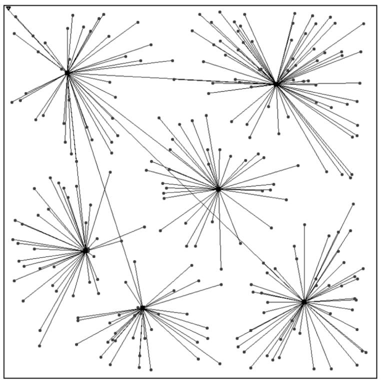

Figure 3: Example of 256 nodes solution.

The installation cost of a cluster head is relatively expensive to the network. The selection of a node as a cluster head must lead to a significant reduction in the propagation cost, in order to be advantageous. Hence, the selection of two or more cluster heads in the same region, generally is not a good choice. This occurs because the decrease in the propagation cost does not pay off the increase in the installation cost. Thus, the installation of just one cluster head per region is generally enough to keep the information flow in the network.

Despite the aleatory generation of all problem instances, it is possible to observe some patterns in the results obtained after optimization. Some regions are more likely to have cluster heads than others. Those observations gave us the idea of applying the divide-and-conquer paradigm and the coevolutionary algorithm to reduce the main problem into smaller sub-problems. If, generally, there are two, one, or no cluster heads per region in the best solutions, we can divide the network into several square areas and divide the main problem into sub-problems, such that each one consists in searching cluster heads for each area, if needed. Therefore, the difficulty of solving the original problem is diluted into the search for local cluster heads in each cell. Each cell is associated with a population of individuals encoding a partial solution for that cell, i.e., the cluster heads in that cell. Individuals in each cell cooperate to produce a complete solution, which is the topology of the whole network.

It is relevant to remark that the basic genetic algorithm is not capable of finding good solutions for this problem as the memetic algorithm is, see [19]. However, the execution time of the memetic algorithm is very high. Applying the divide-and-conquer paradigm, and the cooperative coevolutionary algorithm, it is possible to reach solutions as good as the hybrid ones with less computational effort.

3.2 Fitness computation

In the new problem formulation, the network is divided into a grid of homogeneous cells where each one has a local objective function. Individuals in each cell use this local objective function to compute their fitness values. The local objective function considers the installation cost of the cluster head in the cell and the evaluation of the propagation cost analyzes only the nodes that can be covered by nodes in the cell, whereas in the global objective function all nodes are analyzed.

Figure 4 illustrates the idea of the cell coverage for a given cell C_i, $1 \leq i \leq L$. The gray square represents the cell area. The set of nodes covered by the cell i is denoted by K_i and is given by all nodes in N that are within the radius of at least one node in C_i. The dots in Figure 4 represent the nodes in K_i. Notice that the set of nodes covered by C_i contains some nodes in adjacent cells. We can see that there are sensors within the cell that are not connected to the cluster head of the cell, while there are outer sensors that are connected to the cluster head. Therefore, the cluster head accepts connections from any node in K_i, including nodes in adjacent cells. Hence, during fitness evaluation, each cell computes only the flow from the nodes that are connected to an inner located cluster head, but all the covered nodes are checked.

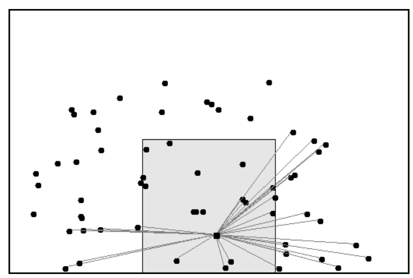

Figure 4: Example of cell coverage.

Let Q_i be the set of inner cluster heads in cell C_i, K_i be the set of nodes covered by cell C_i, such that $|Q_i| \geq 0$, $Q_i \subset H$, $K_i \subset N$, $Q_i \cap K_i = \emptyset$.

The network configuration in each cell can be represented by an integer string of variable size, representing the indices of the nodes that are configured as cluster heads. Each individual $p_{i,t}^{(k)}$ in the sub-population $P_{i,t}$ is represented by an integer vector indicating the cluster heads in the cell C_i at generation t.

We can define the following heuristics in order to eliminate the decision variables q_{ij} in the model (1) when solving the problem with the CCA:

- Given an individual $p_{i,t}^{(k)} \in P_{i,t}$, we obtain the set of nodes in K_i and the set of cluster heads in Q_i: if $s \notin p_{i,t}^{(k)}$, then node s is in K_i, otherwise node s is in Q_i;

- As a general rule, every node $i \in N$ sends its data to the cluster head $j \in H$ with the smallest cost c_{ij}, which is usually the closest one. In this way, we automatically set the corresponding q_{ij} to 1.

The computation of the propagation cost for cell C_i considers only the cost of propagating data from nodes in K_i

that are connected to a given cluster head in Q_i and the cost of routing data from the cluster heads in Q_i to the sink.

With these simplifications, the variables q_{ij} are implicitly calculated from the cluster head allocation provided by an individual $p_{i,t}^{(k)} \in P_{i,t}$, thus simplifying the computation of the propagation cost. The computation of the installation cost is simply given by making $z_s = 1$ if s is in $p_{i,t}^{(k)}$, $s \in N$, and $z_s = 0$ otherwise. Moreover, constraints (2)-(4) can be neglected altogether in the model. We consider only the following constraints: (i) the distance between a node $i \in N$ and its associated cluster head should be smaller than $2r$, in other words, every node $i \in N$ should have a cluster head within its communication range; and (ii) the distance from every cluster head to each other should be smaller than $3r$.

When violations of any of these constraints occur for a given individual, its value of the objective function is penalized accordingly. With the assumptions discussed in this section, we can employ a simplified yet nonlinear model for the problem. This shows the flexibility of CCAs in solving nonlinear integer optimization problems.

3.3 Basic Operators

The first step in the CCA is to set the cells C_i, $i = 1, \ldots, L$, and their sets K_i of covered nodes, based on their coordinates and communication radius. Then, an initial population for each cell is generated by using the new population generator described in the next subsection (subsection 3.4). Each cell has its own subpopulation with μ individuals, each one encoding candidate configurations for the cell. From this initial population, each cell sets its representative individuals. The representative is the best individual that each cell uses to perform the cooperation amongst the other cells. The combination of each individual and the representatives from other subpopulations generates the global solution to the main problem.

Each cell evolves and evaluates its local subpopulation independently from each other, considering the local objective function, as described before, until the cooperation interval is reached. The cooperation interval γ was set to 5 generations. At every $\gamma = 5$ generations, cells communicate their best solution based on current information about the representatives from other cells.

During the cooperation interval, each cell updates its information about the representatives from other cells. The cell containing the cluster head that is the nearest one to the sink is elected as the master cell. It is responsible for evaluating the global solution during the cooperation interval and storing the best individual.

The evolution of each subpopulation is implemented using binary tournament selection for the reproduction, in which two individuals are randomly selected from the current population and compete against each other, and mutation operators specifically designed for the encoding scheme adopted. There are no crossover operators.

The new candidate solutions are produced only by mutation operators. There are three mutation operators with the same mutation rate $\rho_m = 0.1$. The first mutation adds at random a new cluster head from C_i into the vector $p_{i,t}^{(k)}$. The second mutation randomly swaps one cluster head in $p_{i,t}^{(k)}$ for a new cluster head from C_i. The third mutation operator removes at random a cluster head from $p_{i,t}^{(k)}$. These operations allow an individual to have both an empty vector of cluster heads and a vector containing several cluster heads.

3.4 Population Generators

In our previous work [19], the initial population was formed by randomly generated individuals, containing 30% of its nodes set as cluster heads, called 30 Percent Random (30PR). As observed, optimized solutions generally have much fewer cluster heads. The following generators can be devised:

30 Percent Random (30PR): Individuals are generated with 30% of its nodes encoded as cluster heads;

L Overall Random (LOR): Individuals are generated with L nodes selected as cluster heads;

L Cells Random (LCR): Individuals are generated with L nodes selected as cluster heads, but ensuring only one cluster head in each cell;

In this paper we propose a new population generator that is more suitable for the design of WSNs. It is based on the betweenness centrality value [8] of the nodes. The betweenness centrality can be interpreted as a measure of the influence a node has over the spread of information through the network. It can be measured as a fraction of the shortest paths between pairs of vertices in a network that pass through the node. In other words, the betweenness can quantify the importance of a vertex to the network, which can be defined as follows:

$$B_u = \sum_i \sum_j \frac{\sigma(i,u,j)}{\sigma(i,j)}, \tag{8}$$

where $\sigma(i,u,j)$ is the number of shortest paths between vertices i and j that pass through vertex u, $\sigma(i,j)$ is the total number of shortest paths between i and j, and the sum is over all pairs i,j of distinct vertices [6].

First, the betweenness centrality of the nodes in V is measured. Then, a probabilistic operator based on such values is applied to select each cluster head for all cells. This population generator is called L Cells Betweenness Centrality Based (LCBCB). The probability ρ_{Q_i} of a node $u \in C_i$ to be selected as a cluster head can be defined as follows:

$$\rho_{Q_i}(u) = \frac{B_u}{\sum_{k \in K_i} B_k}, \tag{9}$$

In other words, the higher the importance of the vertex u for the covered set K_i, the greater the number of paths in which u is inserted, and the higher the probability $\rho_{Q_i}(u)$ is.

4. SIMULATION RESULTS

4.1 Results of the Population Generators

Table 1 compares the efficiency of all four population generators, considering that the region is divided into a 3×3 grid, such that $L = 9$. The table contains the average, best and worst values of 3300 fitness evaluations, such that 100 individuals were evaluated for each one of 33 random instances, for each network size. These fitness values were normalized, divided by $10^{log_2|V|}$.

Table 1: Performance of the population generator algorithms.

	Worst	Average	Best
64 nodes			
30PR	4728.4 ± 628.5	2761.7 ± 411.0	1041.416 ± 408.1
9OR	1382.1 ± 210.7	589.2 ± 88.0	59.875 ± 48.1
9CR	1146.2 ± 142.9	632.8 ± 87.9	202.716 ± 64.5
9CBCB	572.1 ± 81.1	252.7 ± 70.1	80.701 ± 53.7
128 nodes			
30PR	4896.5 ± 339.6	3716.8 ± 217.0	2573.6 ± 291.9
9OR	362.5 ± 33.6	192.7 ± 10.3	54.2 ± 22.0
9CR	314.0 ± 22.4	210.8 ± 9.3	104.5 ± 21.4
9CBCB	208.4 ± 14.6	123.7 ± 9.1	59.6 ± 11.7
256 nodes			
30PR	3859.2 ± 161.7	3320.6 ± 142.6	2795.0 ± 180.1
9OR	78.7 ± 8.8	43.6 ± 1.9	18.0 ± 4.9
9CR	66.3 ± 3.7	47.3 ± 1.2	29.7 ± 3.9
9CBCB	50.6 ± 4.2	33.8 ± 1.4	20.7 ± 1.8
512 nodes			
30PR	2725.7 ± 56.3	2505.4 ± 55.4	2290.8 ± 71.1
9OR	25.1 ± 3.7	8.5 ± 0.2	3.9 ± 0.7
9CR	18.4 ± 1.6	9.1 ± 0.2	6.3 ± 0.4
9CBCB	16.3 ± 1.4	7.1 ± 0.2	4.8 ± 0.3

Choosing the number of cluster heads based on a fixed percentage of say 30% produces the worst results for all network sizes. The strategy 9CBCB provided, in average, better individuals, which shows the existence of a relationship between high measurements of the betweenness centrality of the cluster heads and the quality of the initial solutions. For all network sizes, the strategy 9OR was better than 9CR for the average and the best values, but again the 9CBCB generator provided better individuals on average. Although the best individuals produced by 9OR are on average slightly better than those produced by 9CBCB, the smaller standard deviation shows the robustness of the 9CBCB generator.

In instances with many nodes, $N = 256$ and $N = 512$, the differences between the results provided by 9OR, 9CR and 9CBCB strategies decrease, but they are all much better than solutions generated by 30PR. This indicates that the partition of the region into 9 cells was very useful for the generation of good individuals. On the other hand, the attenuation of the difference for bigger instances was probably caused by the decrease in the probability of selecting the best cluster heads. However, for all instances, the 9CBCB strategy shows to be considerably better than the previous ones.

4.2 Results of the Coevolutionary Algorithm

The coevolutionary algorithm searches for a WSN configuration with a physical topology that meets the given complex network measures, i.e., a network having a small average path length between every node and the sink and a high clustering coefficient. Thus, when a tree based routing is built over this physical topology, the data traffic is reduced and consequently the energy consumption and the data delivery time are reduced. Initially, we present some general assumptions for the coevolutionary algorithm evaluation:

Simulations: The simulation was performed with the algorithm implemented in Java. The coevolutionary algorithm was executed considering a population of $\mu = 50$ individuals, for 500 generations, with the maximum tolerance of 50 generations without any improvement in the current best solution. The number of necessary simulations is given by [11]:

$$\#rounds = \left(\frac{100\,\xi\,\sigma}{\rho\,\overline{X}}\right)^2, \quad (10)$$

where ξ is a constant of value 1.96, σ is the standard deviation found in the first simulations, $\overline{X}$ is the average of the obtained values and ρ is the percentage of the average that we want to get as deviation, which in this case was 5%. We consider 30 rounds with random topologies, and for each topology we executed the genetic algorithm 33 times and the results are presented with symmetrical asymptotic confidence interval of 95%. The tests are executed in a machine Intel Core i5 2.4GHz with 4GB RAM.

Network topology: The network density is kept constant, the area is $A = \pi\, r^2\, |V|/\delta$, where r is the radius range, $|V|$ is the number of nodes and δ is network density (arbitrarily chosen with the value 8.4791). The nodes started the execution with the same hardware configuration, at the end the cluster head nodes reconfigure their radio frequency based on the infrastructure solution. Thus, the final solution has a heterogeneous WSN.

Resultant configuration: Consider $G = (V, E)$ the initial WSN graph and $G^* = (V, E^*)$ the graph returned by the coevolutionary algorithm. The resultant configuration, used in a real network will be the graph $G' = (V, E')$, where $E' = E \cup E^*$. Thus, the cluster coefficient and the average shortest path length are calculated considering those graphs. Finally, the number of installed cluster heads and the time to compute the results is shown.

Table 2 show the results comparing the Basic Genetic Algorithm (BGA) and Hybrid Genetic Algorithm (HGA), see [19], with the new cooperative coevolutionary algorithm (CCA), considering the variation of the number of nodes as $\{64; 128; 256; 512\}$. The evaluated parameters are: (i) number of installed cluster heads; (ii) the fitness of the solutions, whose values were normalized, divided by $10^{log_2|V|}$, where $|V|$ is the number of nodes; (iii) execution time needed (in seconds) to reach the best solution; and (iv) the convergence in generations. Table 3 compares the results of the initial random geometric graph G and the complex graph G' using two complex network metrics: (i) the cluster coefficient; and (ii) the average shortest path length. These metrics were not directly taken into consideration during the evolutionary process, representing a consequence of the optimization rather than an explicit objective of the problem formulation.

The results presented in Table 2 and Table 3 show that, with the coevolutionary algorithm, it is possible to build a physical topology of the WSN with two specific complex network characteristics, the high cluster coefficient and the low average shortest path length. As we can see, the cluster coefficient of the graph G' is roughly the same value of the original geometric graph G, and the average shortest path length of the graph G' for the networks was reduced, when

Table 2: Comparison of the performance of the previous versions and the new coevolutionary algorithm.

N	Cluster Heads			Fitness$(.10^{log_2 N})$			Time to Best (s)			Convergence (g)		
	BGA	HGA	CCA	BGA	HGA	CCA	BGA	HGA	CCA	BGA	HGA	CCA
64	8	4	4	23.289	20.925	44.165	1.39s	7.63s	0.29s	109.610	11.578	28.315
128	11	6	6	67.317	43.457	51.918	5.15s	3m 53s	2.41s	104.580	16.268	40.386
256	27	6	8	222.932	31.857	36.643	46.73s	1h 13m 19s	20.98s	131.150	39.414	60.207
512	96	9	8	832.056	13.525	14.100	7m 6s	22h 35m 17s	1m 59s	118.830	31.216	46.623

Table 3: Comparison of the considered complex networks metrics.

N	Clustering Coefficient				Average Shortest Path			
	G	BGA	HGA	CCA	G	BGA	HGA	CCA
64	0.73	0.73	0.73	0.73	1.96	1.95	1.94	1.93
128	0.69	0.69	0.70	0.69	2.67	2.61	2.54	2.57
256	0.66	0.65	0.66	0.66	3.67	3.51	3.60	3.40
512	0.63	0.66	0.63	0.65	5.09	4.22	4.92	4.57

compared with the original geometric graph G. Based on this physical topology, a routing algorithm can be used to build the best tree based logical topology. Considering energy consumption and delay, this new logical topology based on complex network measures will always be better than a logical one based on the original geometric graph. This occurs because the number of retransmissions is greatly reduced when cluster head nodes are used.

The quality of the solutions returned by the CCA are as good as the solutions returned by the HGA, except for the instances with 64 nodes. For small network sizes, the solutions returned by the hybrid algorithm did not have the same cluster head location behavior as observed in the other instances. Therefore, the idea of dividing the network into a grid was not very helpful for small instances. However, for the instances with a higher number of nodes, the CCA is, in a sense, better than the other algorithms, because it can provide good physical topologies for the network with the best execution time. The elapsed execution time to reach the best solution of the CCA is lower than for the other algorithms, for all network sizes. This shows that the divide-and-conquer approach was indeed advantageously applied: the main problem could be reduced into smaller problems that are easier to solve, making the cooperative coevolutionary approach very useful for this problem.

On the other hand, the bad performance of the BGA can be justified by the use of inappropriate genetic operators and a superficial exploration of the problem nature. Usually, the topologies provided by the BGA have a high unnecessary number of cluster heads, which are expensive for the whole network. Moreover, the binary codification of the individuals into $|V|$ bits, used in BGA, is inefficient due to the low number of cluster heads that are needed.

Considering the tree based routing aspects over the physical topology, a low average shortest path length avoids, mainly, the data delivery delay. The drawback, in this case, is that when the extra radio is enabled more energy is consumed, but considering the global energy consumption, this approach can actually save energy. The complex network having a smaller value for the average shortest path length provides a logical topology with low energy consumption, because a smaller number of hops will be necessary to send data to the sink. Therefore, there seems to be some relation between low fitness in the BGA and small value for the average shortest path length. But the results of the HGA and CCA show that the reduction of fitness values and the fewer cluster heads causes a small increase in the average shortest path length. The reason for this behavior is that when less cluster heads are allocated in the network, the complex network resembles a geometric network, i.e., there will be fewer edges connecting distant points, leading to a small increase in the average shortest path length. Therefore, the average shortest path length of a complex network with fewer cluster heads, corresponding to the solution found by both the HGA and CCA, tends to approach the value of the average shortest path length of the geometric network. In contrast, excess cluster heads examined in the solutions returned by the BGA caused a reduction in the average shortest path, because there are more alternative paths, however the high installation cost of cluster heads makes them inefficient.

For the instances with 64 nodes, in which the cluster heads of the BGA and HGA solutions are more spatially deployed, the lower average shortest path is found in the CCA solutions. That occurs because the CCA cluster heads are concentrated inside its grid cells, resulting from the betweenness centrality based individuals. For the instances with 128 nodes, the HGA solutions have a high centering location. That explains why its solutions have the lowest average shortest path. However, for the instances with 256 and 512 nodes, despite the good quality of the HGA solutions, the lower number of cluster heads results in a lower number of edges in the set E^*, which are not sufficient to provide a significant reduction of the average shortest path. The BGA solutions have the lowest average shortest path for the instances with 512 nodes, since the excessive insertion of unnecessary cluster heads provides a larger set E^* of edges. The installation of 96 cluster heads produces a high installation cost, which is too expensive for the network and can not be deployed in practice. Hence, again, the CCA solutions present themselves as the best option.

5. CONCLUSIONS AND FUTURE WORK

This work presented a cooperative coevolutionary algorithm for fast design of the physical topology in WSN. The goal was to produce a physical topology that presents a high clustering coefficient and a small average shortest path length, which are two independent metrics from the complex network theory for quantifying structural features of the network. This physical topology could be used to improve the tree based routing in order to minimize power consumption and delay. For networks with hundreds of nodes, for instance the network with 512 nodes, the cooperative coevolutionary algorithm was satisfactory, obtaining a physical topology that satisfies the complex network characteristics.

The results showed that the cooperative coevolutionary algorithm can find a WSN design with two specific complex network characteristics. This was highlighted in our results that showed that the cluster coefficient of the resultant graph is the same or slightly higher when compared to the original geometric graph, and the average shortest path length of the resultant graph, in our specific scenario, was reduced when compared to the original geometric graph. The coevolutionary algorithm was able to achieve high quality solutions with the smallest elapsed time, representing configurations with few nodes installed as cluster heads. That means, in practice, that the CCA algorithm is the most feasible, compared with the previous ones.

This complex network strategy is important in WSNs because, when the tree based routing is built over this physical topology, a high cluster coefficient avoids the data delivery delay and unnecessary energy consumption by concentrating the data sensing in a given cluster head. Interferences and link layer processing are avoided when two radios with different communication frequencies are used in the cluster heads. Again, when the tree based routing is built over this physical topology, the low average shortest path length avoids, mainly, the data delivery delay but more "local" energy is consumed, because the extra radio has its communication frequency increased. This discussion shows the truthfulness of the Main hypothesis presented in Section 2.

The coevolutionary model developed in this paper is an important step towards a parallel implementation and even a distributed algorithm for the design of WSNs, which represent important steps in future work.

Acknowledges

This work is partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under the grant number 477946/2010-0 and the Research Foundation of the State of Minas Gerais (FAPEMIG) under the grant number CEX-APQ-00577-09.

6. REFERENCES

[1] K. Akkaya and M. Younis. A survey of routing protocols in wireless sensor networks. *Ad Hoc Network*, 3(3):325–349, May 2005.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[3] P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In *5th Int. Conf. Genetic Algorithms*, pages 264–270, 1993.

[4] D. J. Baker and A. Ephremides. The architecture organization of a mobile radio network via a distributed algorithm. *IEEE Trans. on Communications*, 29(11):1694–1701, November 1981.

[5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[6] L. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. In *Advances in Physics*, volume 56, pages 167–242, August 2007.

[7] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *26th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, volume 4, pages 2033–2036, Salt Lake City, Utah, USA, June 2001. IEEE Computer Society.

[8] L. C. Freeman. A set of measures of centrality based on betweenness. In *Sociometry*, volume 40, pages 35–41, 1977.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless sensor networks. *Proc. IEEE Hawaii Intl Conf. System Science*, pages 1–10, 2002.

[10] S. F. Hwang, Y. Y. Su, Y. Y. Lin, , and C. R. Dow. A cluster-based coverage-preserved node scheduling scheme in wireless sensor networks. *Proc. 3rd Ann. Intl Conf. Mobile and Ubiquitous Systems: Networking and Services*, pages 1–7, 2006.

[11] R. K. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* John Wiley & Sons, April 1991.

[12] Y. Li, M. T. Thai, and W. Wu. Wireless sensor networks and applications. *New York: Springer*, 2008.

[13] Y. Liu, X. Yao, Q. Zhao, , and T. Higuchi. Scaling up fast evolutionary programming with cooperative coevolution. In *Parallel Problem Solving from Nature VIII Conference*, pages 882–891, 2001.

[14] M. Maróti. Directed flood-routing framework for wireless sensor networks. In *5th ACM/IFIP/USENIX International Conference on Middleware (Middleware'04)*, volume 78, pages 99–114, Toronto, Ontario, Canada, October 2004. ACM.

[15] M. A. Potter and K. A. D. Jong. A cooperative coevolutionary approach to function optimization. In *Parallel Problem Solving from Nature III Conference*, pages 249–257, 1994.

[16] W. Qiu, E. Skafidas, and P. Hao. Enhanced tree routing for wireless sensor networks. *Ad Hoc Networks*, 7(3):638–650, May 2009.

[17] C. D. Rosin and R. K. Belew. New methods for competitive coevolution. In *Evolutionary Computing*, volume 5, pages 1–29, 1997.

[18] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[19] A. S. Ruela, R. S. Cabral, A. L. L. Aquino, and F. G. Guimarães. Memetic and evolutionary design of wireless sensor networks based on complex network characteristics. *International Journal of Natural Computing Research*, 1(2):33–53, April 2010.

[20] J. H. Ryu, S. Song, and D. H. Cho. New clustering schemes for energy conservation in two-tiered mobile ad hoc networks. *IEEE Trans. on Vehicular Technology*, 51(6), November 2002.

[21] G. Sharma and R. Mazumdar. Hybrid sensor networks: A small world. In *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobHoc'05)*, pages 366–377, Urbana-Champaign, Illinois, USA, May 2005. ACM.

[22] K. Sohraby, D. Minoli, and T. Znati. Wireless sensor networks: Technology, protocols, and applications. *Hoboken, NJ: Wiley-Interscience*, 2007.