# An Energy-Efficient Clustering Algorithm for Wireless Sensor Networks

**Victor de Oliveira Matos, José Elias C. Arroyo, André Gustavo dos Santos and Luciana B. Gonçalves**

Computer Science Department - Federal University of Viçosa – Viçosa - MG - Brazil

**Summary**
In wireless sensors networks (WSNs) the efficient use of the sensors' energy is a key point to extend the network lifetime and has been the center of attention by many researchers. Clustering formation is an important technique used to reduce the sensor's energy consumption. However, finding an optimal clustering in WSNs is a NP-Hard problem, thus efficient heuristics are needed to find good clustering in reasonable time. In this work we propose and analyze a Greedy Randomized Adaptive Search Procedure (GRASP) coupled with Path Relinking intensification to obtain near-optimal clustering. We develop a clustering protocol to simulate the clusters formation and data transmission. The good performance of our protocol is compared with the well-known clustering protocols for WSNs, LEACH and LEACH-C. The comparison results and the statistical analyses reveal that the GRASP algorithm determines better clusters extending the network lifetime.

***Key words:***
*Clustering, Wireless Sensor Network, Heuristics, Combinatorial Optimization.*

## 1. Introduction

Wireless sensors network consists of a large number of sensor nodes responsible for monitoring some kind of physical or environmental phenomena such as movement, sound and pollution. Its applications range from military tactics and security to environment monitoring. These sensors usually have a microprocessor, one or more sensing devices, limited memory and battery and a wireless transceiver [1]. The sensors can be deployed in various outdoors environments, such as forests, rivers and even in inhospitable or inaccessible environments such as battlefields, deserts and oceans [19]. Therefore it may be impossible or unprofitable to replace their battery or supplying additional energy. When a sensor runs out of battery, it became inactive, leaving a hole in the monitored area. As the sensors can be used as relay nodes in the network, the depletion of a node can cause the whole network to fall apart. Thus an efficient use of the sensors energy is very important to maximize the network lifetime [4]. Clustering protocols have emerged as a popular and efficient way to save energy in WSNs.

Clustering is a technique to efficiently manage the network energy consumption by reducing the transmission range of the sensors. In this technique, the nodes of the network are divided into groups known as clusters and each group has a leader node called cluster head, responsible to manage the group communication with the sink node (base station). So, the sensors no longer transmits directly to the sink, they send the collected information to the cluster head that aggregates the whole group message before sending it to the sink. Figure 1 shows an example of a network configuration with thirty two sensors, of which five of them acting as cluster heads. In this example we can see that the member nodes (regular sensors) transmit the sensed data to their cluster heads and, each cluster head transmit its packet to the base station.

As the data collected by neighboring sensor nodes is usually highly correlated, redundant information is very common [1]. Data aggregation techniques can be used to reduce a large amount of similar data into a smaller set of data that maintains the effective information. This reduces the amount of data needed to be transmitted [13].

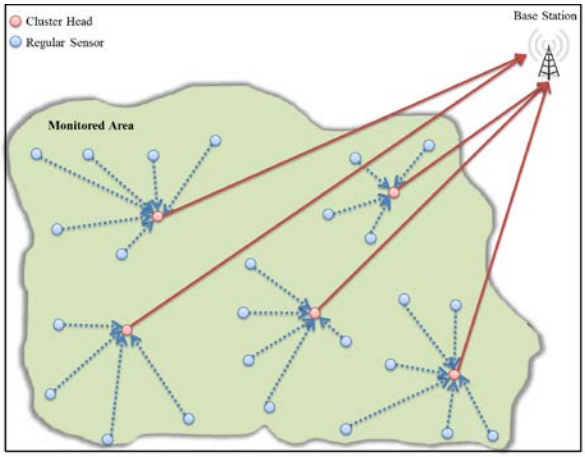


Fig. 1 - An example of a clustering scheme

Energy is consumed when a sensor send or receive messages and when it performs the data aggregation. Therefore, the cluster head consumes much more energy comparing with common nodes that only do the monitoring and sending information. If the cluster head dies, the whole cluster becomes inactive and the collected data by the cluster members will not be sent to the sink. Thus it is

Manuscript received October 5, 2012
Manuscript revised October 20, 2012

important to do a cluster head rotation in a way to avoid a premature death of a sensor node. This rotation can be done from time to time or after some amount of data has been transferred by the network.

Several cluster based protocols have been proposed in clustering technique literature. One of the well-known clustering protocols called LEACH (Low Energy Adaptive Clustering Hierarchy) has been proposed in [7] to maximize the network lifetime. This protocol includes distributed cluster formation and the clusters are formed randomly. The nodes elect themselves as cluster heads with some probability. The algorithm is run periodically and the probability of becoming a cluster head for each period is chosen to ensure that every node becomes a cluster head at least once within 1/p periods, where p is a predetermined percentage of cluster heads. LEACH protocol organizes its operation into rounds. Each round consists of two phases a setup phase and a transmission phase. In the setup phase the clustering algorithm determines the clusters (cluster heads nodes are determined). In the transmission phase a data communication process is executed (the nodes send the collected data to the cluster head and on to the sink). These two phases are repeated by a number of periods (rounds), while the network is alive (there are percentage of alive sensors). After the energy dissipated in a given node reached a predefined threshold, that node is considered dead for the remainder of the rounds. LEACH provides significant energy savings and prolonged network lifetime over conventional multihop routing schemes, such as the Minimum Transmission Energy routing protocol [7]. However, LEACH does not guarantee that the desired number of cluster heads is selected and cluster heads are not evenly positioned across the network. A further improvement of this protocol known as LEACH-C was proposed in [8]. In LEACHC, the cluster formation is done at the beginning of each round using a Centralized algorithm by the base station. The base station uses the information received from each node during the setup phase to find a predetermined number of cluster heads and configures the network into clusters. The clusters are then chosen to minimize the energy required for non-cluster head nodes to transmit their data to their respective cluster heads. Results in [8] have shown that the overall performance of LEACH-C is better than LEACH due to improved cluster formation by the base station. Moreover, the number of cluster heads in each round of LEACH-C is equal to the desired optimal value, whereas for LEACH the number of cluster heads varies from round to round due to the lack of global coordination among the nodes.

Another clustering protocol which aims to enhance the network lifetime is presented in [10]. Power -Efficient Gathering in Sensor Information Systems uses a greedy algorithm to organize nodes into a chain, so that each node transmits and receives from only one of its neighbors. In each round, a randomly chosen node from the chain will transmit the aggregated data to the base station and reduce the number of nodes that communicate directly with the base station. In [12], an approach called Base station Controlled Dynamic Protocol is proposed which produces clusters of equal size to avoid cluster head overload and to ensure similar power dissipation among the nodes.

The application of metaheuristic algorithms, such as Genetic Algorithms and Particle Swarm Optimization, to solve the problem of sensor network clustering has been proposed in the literature [9] [17] [18] [2].

In this work, we used a fixed number of nodes to be cluster heads depending on the amount of alive nodes, a single-hop communication intra-clusters and multi-hop among the cluster heads to reach the sink. The cluster heads are common nodes, selected from all alive nodes. A centralized algorithm based on GRASP [5] was developed to solve the clustering problem in wireless sensor network. The GRASP [5] algorithm was used to determine the list of cluster heads. The algorithm consists of three phases. On the construction phase, a valid solution is generated with a randomized greedy algorithm. Then, a local search procedure is used to improve the solution obtained in the construction phase. The third phase is an intensification strategy using the Path Relinking technique, which combines two solutions to find better ones. The proposed algorithm is tested with a set of instances, ranging from 50 to 300 nodes and is compared with clustering protocols from the literature. To the best of our knowledge, this is the first application of GRASP heuristic to obtain energy-efficient clusters in WSNs.

This paper is structured as follows: section 2 presents the description of the clustering problem and used system model. Section 3 provides a detailed description of the proposed clustering algorithm. Section 4 presents the generation mode of the used instances and the calibration of the proposed algorithm. Results of computational experiments to evaluate the performance of the proposed algorithm are reported in Section 5. Finally, Section 6 provides the concluding remarks.

## 2. Clustering Problem and System Model

The clustering problem in WSN can be defined as follows: Given a set G of n nodes and a sink node (or base station) s, randomly positioned in the monitoring area. The problem is to find a set of nch nodes to be cluster heads that minimizes the amount of energy for the non-cluster head nodes (regular sensors) to transmit their data to the cluster head, by minimizing the total sum of squared distances between all the non-cluster head nodes and the closest cluster head and between each cluster head to the sink

node. This problem is a NP-hard combinatorial optimization problem, thus it is very unlikely to develop an efficient algorithm to solve it exactly [3]. In this work we adopt an area with no obstacles and consider squared distances d2. Therefore, the objective function to be minimized is defined as:

$$f = \sum_{i=1}^{n} d(i, ch)^2 + \sum_{k=1}^{n_{ch}} d(ch_k, s)^2 \quad (1)$$

Where, $n_{ch}$ is the number of cluster heads, $d(i, ch)$ is the distance between node $i$ and its nearest cluster head $ch$, and $d(ch_k, s)$ is the distance from the cluster head $ch_k$ to the sink node $s$. As in [7], the simulations are done with $n_{ch}$ = 5% of the alive nodes.

### 2.1. System Model

In this work, the energy model for the sensors is based on the first order radio model as used in [7] and [8]. In this model the sensors dissipates energy for transmitting, receiving and aggregating messages. The transmitter dissipates energy by using the radio and the amplifier. Receiving only dissipates energy by using the radio, as shown in Figure 2. The radio can control the transmission power to use only the necessary energy to reach the intended node.

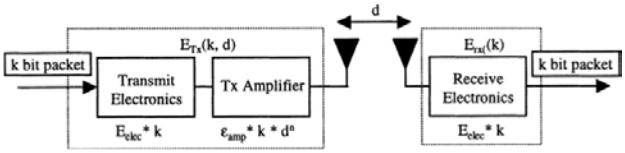


Fig. 2 - The Radio Model [7]

Using the radio electronics, consumes $E_{elec}$ = 50nJ/bit and the amplifier consumes $\varepsilon_{amp}$ = 10pJ/bit/m2. To *transmit* a message of $k$ bits to a distance $d$ and to *receive* a message of $k$ bits the radio expends, respectively, the following energies:

$$E_{tx}(k, d) = E_{elec} \times k + \varepsilon_{amp} \times k \times d^2 \quad (2)$$

$$E_{rx}(k) = E_{elec} \times k \quad (3)$$

The energy to perform the data aggregation is set as $E_{DA}$ = 5nJ/bit. In a cluster with $h$ nodes, each node collects and sends a package of size $k$ bits to the cluster head. Therefore, we assumed that the aggregation algorithm used in the simulation can compress the overall collected information from each cluster into a single package of size $k$, regardless of the number of nodes in that cluster.

We consider a sensor network model similar to those used in [8], with the following properties:

- Each sensor constantly monitor the ambient and always has data to send to the sink node;
- The sink node is randomly deployed in the sensing area and has unlimited energy;
- The sensors are stationary and starts with the same energy;
- The sensors can vary the transmitting power to consumes only the necessary energy to reach the destiny sensor;
- All sensors can operate in cluster head mode and sensing mode;
- There is no distance limits; this way, every sensor can reach all other nodes.

## 3. GRASP algorithm for Cluster Formation

The Greedy Randomized Adaptive Search Procedure, proposed by [5] is a well-known technique used to solve several combinatorial optimization problems. GRASP is a multistart (iterative) method that consists of two phases: a construction phase and an improvement (local search) phase. The construction phase builds solution step by step, adding elements to a partial solution. The element to add is selected according to a greedy function which is dynamically adapted as the solution is built. However, the selection is not deterministic, but subjected to a randomization process. When a feasible solution has been built, its neighborhood is explored in a local search phase until a local optimum is found. The best solution produced from all GRASP iterations is returned as the output.

In this paper, an adaptation of the GRASP algorithm is used to solve the clustering problem (cluster formation) in WSN. We add an intensification phase to the GRASP algorithm. This intensification is based on Path Relinking technique which combines the solution returned by the local search with good quality solutions stored in a set Pool. The proposed algorithm is called GRASP+PR and its pseudo code is presented in Algorithm 1. The algorithm has two parameters, *max_it* and α. *max_it* is the number of iterations without improvement which is adopted as the stop condition of the algorithm and the parameter is used in the local search procedure

At the beginning of the algorithm, a set *LC* of the alive nodes that can be cluster head is determined. A node will be a cluster head candidate only if it has energy above an estimated limit $E_{min}$. To estimate the energy consumed by a cluster head in one round, we considered that the clusters have the same amount of nodes, therefore, all the cluster heads consume the same amount of energy, changing only their distances to the sink ($d_{sink}$). The energy threshold is defined as:

$$E_{\min} = \frac{RL/k}{n_{ch}} \times E_{rx}(k) \times E_{DA} \times E_{tx}(k, d_{sink}) \quad (4)$$

where $RL$ is the amount of data specified to finish one round and $k$ is the packet size.

In the next subsections, we described each phase of the GRASP+PR algorithm presented in this paper.

**Algorithm 1 GRASP+PR** (*max_it, α*)
$f(CH_b) := \infty$;
*Pool* := ∅;
*It* := 0;
*LC* := set of potential nodes to be cluster head;
**while** *it* < *max_it* **do**
    *CH* := *SampleGreedy*(*LC*);
    *CH* := *LocalSearch*(*CH*, *α*, *LC*);
    *addSolutionPool*(*CH*);
    **for each** *CH'* ∈ *Pool* **do**
        **if** *CH* ≠ *CH'* **then**
            $CH_{pr}$ := *PathRelinking*(*CH*, *CH'*);
            **if** $f(CH_{pr}) < f(CH)$ **then**
                $CH := CH_{pr}$;
                *addSolutionPool*(*CH*);
            **end-if**;
        **end-if**;
    **end-for**;
    **if** $f(CH) < f(CH_b)$ **then**
        $CH_b := CH$;
        *it* := 0;
    **else**
        *it* := *it* + 1;
    **end-if**
**end-while**
**return** $CH_b$;

## 3.1 Solution Representation and Construction Phase

A solution of the clustering problem is represented by two vectors, *CH* and *I*. *CH* is of size $n_{ch}$ and it contains the cluster head nodes in the current round. Indexes of the vector *I* (of size *n*) represent node IDs and the values of respective cells are their cluster head ID. In this structure, for a node which is chosen as cluster head, the value of respective cell is its own ID. By the use of such structure, we can assign member nodes to the cluster heads. Figure 3 shows an example of a solution representation with $n = 10$ and $n_{ch} = 3$. In this example we suppose that all nodes are candidates to be cluster heads i.e. $LC = \{1, 2, \ldots, 10\}$.

In the construction phase a solution is constructed by using the sample greedy algorithm proposed in [16] for the p-median problem. This algorithm starts with an empty solution ($CH = \emptyset$) and adds nodes to *CH* one at a time. In each iteration, instead of selecting the best among all nodes of the candidate list *LC*, it only considers a sample set *Q* ($\subset LC$) of the possible nodes (chosen uniformly at random). The most profitable among those nodes is selected and added to *CH*. The size of the sample set ($q = |Q|$), is defined as in [16]: $q = \left\lceil \log_2\left(n / n_{ch}\right) \right\rceil$. The idea is to make *q* small enough so as to significantly reduce the running time of the algorithm (when compared to the pure greedy one) and to ensure a fair degree of randomization. In Algorithm 2 the pseudo code of the used sample greedy algorithm is presented.

| *CH* | 4 | 1 | 6 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *I* | 1 | 1 | 6 | 4 | 1 | 6 | 1 | 6 | 1 | 1 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Fig. 3 - The Solution Representation

**Algorithm 2** ***SampleGreedy*** (*LC*)
*CH* := ∅;
**while** $|CH| < n_{ch}$ **do**
    *Q* := set of *q* nodes selected randomly from *LC*;
    *ch* := select the best node of *Q* to be cluster head;
    $CH := CH \cup \{ch\}$
**end-while**
**return** *CH*;

## 3.2 Local Search

Local search is an improving method based on neighborhood search. This method receives a solution *CH* obtained by the construction phase and performs cluster head swaps on this solution. The *swap* movement determines a new solution (neighbor solution) by removing from *CH* a cluster head *ch* and inserting in its place a node $c \notin CH$. Swap movements are commonly used in local search procedures for the *p*-median problems and have been shown to be effective [16]. To reduce the number of neighbor solutions, we define a subset *C* of candidate nodes that can be inserted in the solution (nodes that can be swapped with *ch*). This set is formed by α percent of total nodes in *LC* – *CH*.

A node $ch \in CH$ is swapped with all the nodes $c \in C$. From these swaps the best solution *CH** is chosen. If solution *CH** is better than the initial solution *CH*, this solution is updated as the current solution.

The local search algorithm tests the removal of all the nodes $ch \in CH$. The removal of these nodes is done in greedy way, that is, first it is removed the worst node, in other words, the node that improves less the solution.

If the removal of a node *ch* do not improve the current solution, the removal of this node is prohibited (this node is added to a set *R*). Whenever the current solution *CH* is updated, the set *R* is emptied. This means that a removed node can be inserted in the solution and removed again in future iterations. In Algorithm 3 is presented the pseudo code of the proposed local search procedure. Note that the algorithm ends when $|R| = n_{ch}$, i.e. when all cluster heads of *CH* are prohibited to be removed.

**Algorithm 3** ***LocalSearch*** (*CH*, α, *LC*)
*R* := ∅; {list of removed nodes}
**while** |*R*| < $n_{ch}$ **do**
  *ch* := select the worst node *ch* ∈ *CH* such that *ch* ∉ *R*;
  *C* := select randomly α × |*LC* - *CH*| different nodes from *LC* – *CH*;
  *CH** := *CH*;
  **for** each node *c* ∈ *C* **do**
    *CH'* := solution obtained from *CH* by swapping nodes *ch* and *c*;
    **if** *f*(*CH'*) < *f*(*CH**) **then**
      *improved* := **true**;
      *CH** := *CH'*;
    **end-if;**
  **end-for**
  **if** *improved* **then**
    *CH* := *CH**;
    *R* := ∅;
  **else**
    *R* := *R* ∪ {*ch*};
  **end-if**
**end-while**
**return** *CH*;

### 3.3 Intensification Phase: Path Relinking

In this work, the Path Relinking (PR) technique is used to search new good quality solutions. PR was originally proposed by Glover [6] as a mechanism to combine intensification and diversification by exploring trajectories connecting high quality (elite) solutions previously produced during the search. The PR needs a pair of solutions, say $CH_o$ (origin solution) and $CH_g$ (guiding solution), $CH_o \neq CH_g$. Solution $CH_o$ is the solution returned by the local search and the guiding solution $CH_g$ is a solution selected at random from a set *Pool* of elite solutions.

The PR procedure starts with $CH_o$ and gradually transforms it into the other solution $CH_g$ by inserting nodes from $CH_g - CH_o$ and removing nodes from $CH_o - CH_g$ (i.e. a path that links $CH_o$ to $CH_g$ is generated by swap movements). The total number of stages made (to transform $CH_o$ into $CH_g$) is $|CH_g - CH_o|$, which is equal to $|CH_o - CH_g|$ (*symmetric difference* between $CH_o$ and $CH_g$). In each stage, the "best" solution obtained is chosen. Solutions less similar than the origin solution and more similar than the guide one are gradually obtained in each step. The PR procedure returns the best solution obtained in the path from $CH_o$ to $CH_g$. In Figure 4 is shown an example in which the guide solution is founded in three stages from the origin solution.

The set *Pool* of elite solutions stores at most $p_{size}$ solutions. This set is updated by the procedure *addSolutionPool*(*CH*), where *CH* is a good solution obtained in each iteration of the GRASP+PR algorithm. This procedure checks if the elite set is full (|*Pool*| = $p_{size}$), *CH* ∉ *Pool* and *CH* has better quality than the worst elite solution in *Pool*. If these conditions are satisfied, then the solution *CH* is added to *Pool*, replacing some elite solution. Among all elite solutions having objective function no better than that of *CH*, the solution *CH'* most similar to *CH* is selected to be removed from the elite set. The similarity between two solutions is determined by their symmetric difference. If the elite set is not full (|*Pool*| < $p_{size}$), a solution *CH* is simply added to the elite set if *CH* ∉ *Pool*.

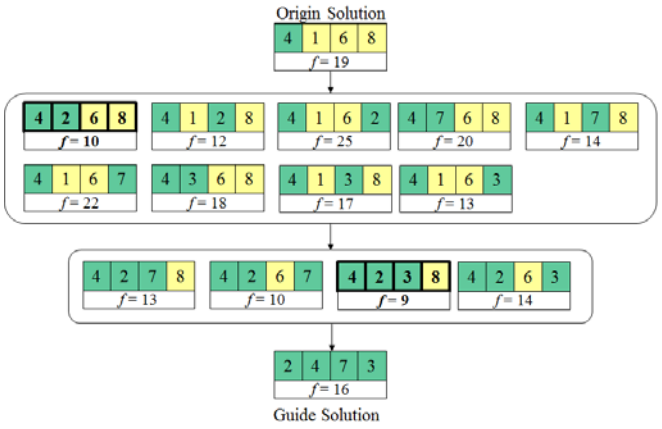


Fig. 4 - Example of Path Relinking

## 4. Benchmark of Instances and Parameters of the GRASP+PR Algorithm

The simulator was developed in C++ and was executed on an Intel Core 2 Quad 3.2 GHz with 3 GB of RAM running under Ubuntu 10.04, 64 bits OS.

To evaluate the performance of the proposed algorithm GRASP+PR, computational experiments are performed on randomly generated instances of the problem. We generated instances with *n* = 50, 60, 70, 80, 90, 100, 125, 150, 200 and 300 sensors. All nodes including the sink node are uniformly deployed within a 100 × 100 area. Each node starts with the same amount of energy $e_i$ = 1J. For each number of sensors (*n*), 10 different instances were created. Therefore, a total of 10 × 10 instances were generated. The instances are ordered according to the number of nodes and numbered from 1 to 100. So, instances 1-10,...,91-100 represent instances with 50,...,300 nodes, respectively.

The proposed GRASP+PR algorithm has three parameters to be adjusted experimentally. These parameters are: the maximum number of iterations without improvement of the best solution (stop condition) (*max_it*), the parameter used in the local search procedure (α) and the size of the elite set *Pool* ($p_{size}$) used in the PR intensification. In this work,

we firstly analyze the parameter of the local search procedure. For this, we run the GRASP algorithm without PR intensification. We test the following values for the parameter α: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9 and 1.0. The parameter *max_it* was set to 40. Note that, there are 10 versions of the GRASP algorithm, one for each value of α. We run five replicates of each version and the average solution (average objective function) among the five runs is chosen as the final output of the algorithm. The obtained results are analyzed by using the Relative Percentage Deviation (RPD) which is computed in the following way:

$$RPD\% = 100 \times \frac{f_{algorithmn} - f_{best}}{f_{best}} \quad (5)$$

where $f_{algorithm}$ is the average objective function value obtained by a version of the algorithm, $f_{best}$ is the best objective function value obtained among all the versions. In order to validate the results, an Analysis of Variance (ANOVA) [11] is applied to check if the observed differences are statistically significant. This analysis has been carried out employing the RPD measure as response variable. The ANOVA analysis indicates that these measures are statistically different for the different values of the parameter. We also carry out a multiple comparison test of Tukey in order to verify for what values of the results are statistically significant.

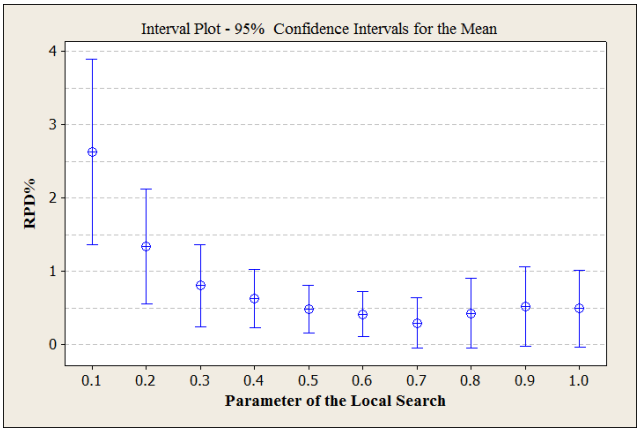


Fig. 5 - Interval Plot - Analysis of the parameter α

Figure 5 depicts the intervals plots with 95% confidence intervals for the mean. We can note that, with α = 0.1 and α = 0.5, the results of the GRASP algorithm are statistically different and the algorithm shows the best performance by using α = 0.5. The results of the algorithm are not statistically different for α = 0.2, …,1.0 (confidence intervals are overlapped). In Figure 5 we can see that the lowest average RPD was obtained with α = 0.7. In order to reduce the computational time of the algorithm, in the following experiments we use α = 0.5 (that is, 50% of the neighbor solutions are analyzed). In the local search procedure, the strategy of scanning many neighbor solutions usually consumes the largest part of the algorithm. After set the parameter α, we analyze the GRASP+PR algorithm by varying the parameters $p_{size}$ (pool size) and *max_it* (GRASP stopping condition). For these parameters we tested the combination of the following values: $p_{size} \in \{2, 5, 10, 15, 20\}$ and *max_it* ∈ {2, 5, 10, 20, 40}. Note that, there are 16 versions or configurations of the GRASP+PR algorithm, one for each combination of $p_{size}$ and *max_it*. The ANOVA analysis indicates that the results are statistically different. Figure 6 shows the intervals plots with 95% confidence intervals for the mean. We can see that for *max_it* = 5 and *max_it* = 40, the results of the GRASP+PR algorithm are statistically different and the algorithm obtains the best results by using *max_it* = 40. We can also see that the lowest average RPD was obtained with the combination $p_{size}$ = 20 and *max_it* = 40. In order to reduce the computational time of the GRASP+PR algorithm, in the following experiments we opt $p_{size}$ = 20 and *max_it* = 20.

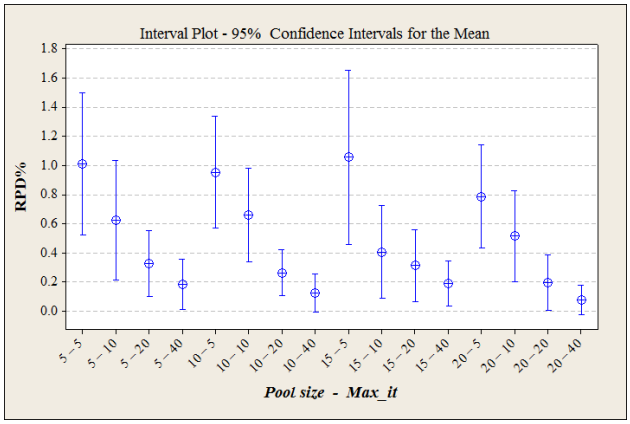


Fig. 6 - Analysis of parameters Psize and max_it

Table 1 lists the simulation parameters and the algorithm parameters adopted in evaluation.

Table 1: Simulation and Algorithm Parameters

| | |
|---|---|
| Initial energy of the sensors ($e_i$) | 1J |
| Radio energy consumption ($E_{elec}$) | 50nJ/bit |
| Amplifier energy consumption ($\varepsilon_{amp}$) | 10pJ/bit/m² |
| Data aggregation energy consumption ($E_{DA}$) | 5nJ/bit |
| Number of cluster heads ($n_{ch}$) | 5% of the alive nodes |
| Local search parameter (α) | 0.5 |
| GRASP Stop condition (*max_it*) | 20 iterations |
| Size of the elite set *Pool* ($p_{size}$) | 20 solutions |

## 5. Computational Results

The implemented clustering protocol operates in rounds, where each round begins with a Setup Phase in which clusters are formed and it is followed by a Transmission Phase in which we used a similar approach as in [7]. Figure 7 shows the flowchart of the implemented clustering protocol. The Setup and Transmission phases are executed until too few nodes are alive to transmit data to the sink. In this work, a simulation ends when the number of alive nodes is less than $0.05 \times n$, where $n$ is the total number of nodes of the initial network.

The effectiveness of GRASP+PR, used at the Setup Phase (clusters formation), is compared with LEACH and LEACH-C protocols. The algorithms LEACH and LEACH-C were also coded in C++ and were executed on the same computer. We also tested the performance of the GRASP algorithm (i.e. without the Path Relinking intensification). For each instance, the algorithms are executed five times and the average solution among the five runs is chosen as the final output.

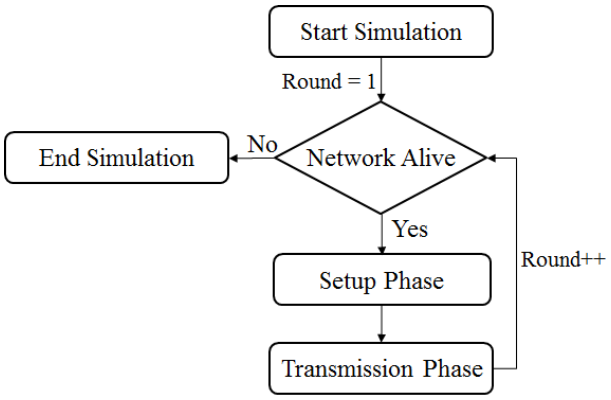


Figure 7 - Network Simulation Flowchart

In this work we use same criteria to evaluate the clusters formation algorithms (GRASP, GRASP+PR, LEACH and LEACH-C), they are: solution cost (objective function), network energy consumption, the number of transmissions in the network and network longevity (network lifetime).

To compare the algorithms by the solution cost, the network simulation is executed for five rounds and the average solution cost among the five rounds is chosen as the output of the algorithm. Table 2 shows the average value of the objective function for each algorithm. The instances were grouped according to the number of sensors. We can see that the proposed GRASP and GRASP+PR achieve considerable lower values when compared with the other algorithms.

The algorithms are also compared by the RPD which is computed regarding the solution cost. The graphic depicted in Figure 8 shows the comparison of the RPD values related to the solution costs for all the 100 instances tested. We can see that LEACH algorithm presents the worst RPD values and higher variance when compared with the other algorithms. This was expected because it has a random nature. LEACH-C proved to be better than LEACH but as the size of the network grows, the solution cost get worse.

Table 2 – Average of the objective function value

| *n* | LEACH | LEACH_C | GRASP | GRASP+PR |
|---|---|---|---|---|
| 50 | 77531.2 | 42631.4 | 35930.8 | 35874.9 |
| 60 | 68603.1 | 40292.7 | 29478.0 | 29315.8 |
| 70 | 84834.1 | 47847.0 | 35628.7 | 35315.0 |
| 80 | 82963.5 | 48517.6 | 33381.6 | 33032.1 |
| 90 | 90387.6 | 51718.4 | 36372.4 | 36102.1 |
| 100 | 81391.6 | 48638.0 | 33212.1 | 32614.0 |
| 125 | 94281.1 | 59845.9 | 35513.2 | 34846.1 |
| 150 | 105444.5 | 68279.9 | 37685.6 | 36836.0 |
| 200 | 110980.6 | 77835.9 | 36990.1 | 35825.2 |
| 300 | 123083.7 | 93090.2 | 36883.1 | 35507.8 |
| **Total Average** | **91950.1** | **57869.7** | **35107.5** | **34526.9** |

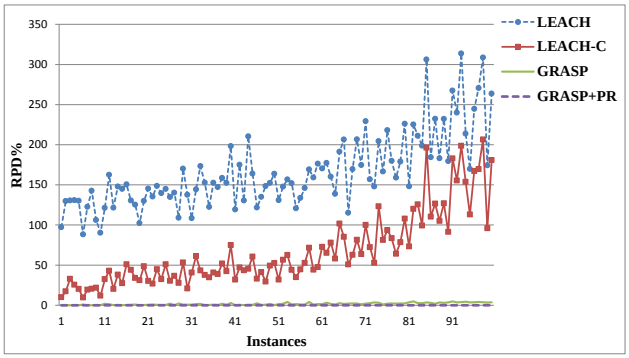


Fig. 8 - Relative percentage deviation (RPD) of the algorithms.

GRASP and GRASP+PR provide better results than LEACH and LEACH-C for all the instances tested. For small instances, GRASP obtains the same solutions as GRASP+PR. However as the number of sensors increases, the difference between the two algorithms becomes higher.

The ANOVA analysis indicates that there is statistically significant difference between the algorithms GRASP and GRASP+PR. Figure 2 depicts the means plots and Tukey confidence intervals with a 95% confidence level. These intervals help to find where the significant difference is. If there is no overlap between the confidence intervals, a statistically significant difference can be assumed. The GRASP+PR algorithm shows no variation when compared with its simple version. The Path Relinking intensification used in GRASP+PR proved to be an efficient strategy to achieve even better solutions. Thus, the GRASP+PR algorithm is statistically better (regarding the solution cost) than the simple GRASP.

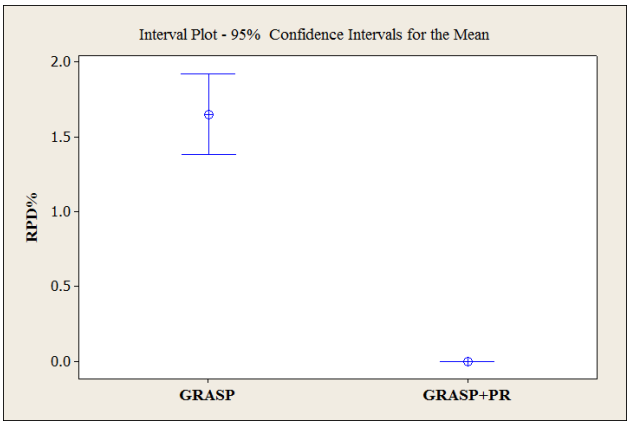


Fig. 9 - GRASP and GRASP+PR solution cost comparison

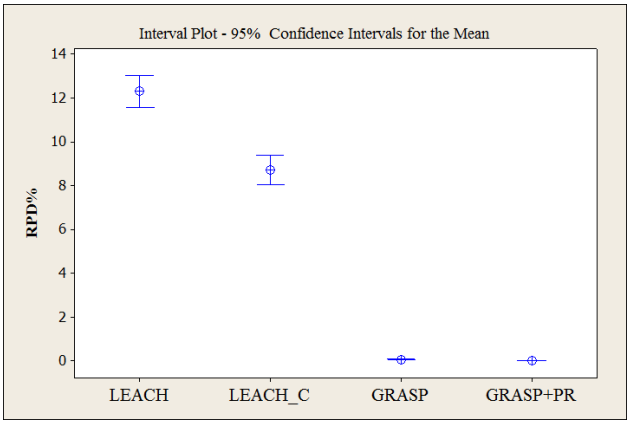


Figure 10 - Energy consumption interval plot

As the objective function value uses directly the transmission cost to be calculated, better solutions will get lower energy consumption. Similarly, we analyze the results of the ANOVA now with the four algorithms. Figure 10 shows the means plots and Tukey confidence intervals with a 95% confidence level. Again, LEACH and LEACH-C showed the worst values of RPD. GRASP and GRASP+PR algorithms are not statistically different (considering energy consumption), that is, on average, the behavior of these two algorithms is the same. However, our algorithms are statistically different and superior by comparison with the algorithms LEACH and LEACH-C.

A network that consumes little energy remains alive for more rounds. For an instance with 100 nodes, Figure 11 shows the number of rounds with respect to the network total remainder energy, for the different clustering algorithms. As clearly noted from this Figure, GRASP+PR clustering algorithm extends the lifetime of the network much more significantly than LEACH and LEACH-C methods. Networks obtained by LEACH algorithm consume considerably more energy than the networks found by algorithms LEACH-C, GRASP and GRASP+PR. These results also show that the algorithms GRASP and GRASP+PR have very similar energy consumption, however, GRASP+PR algorithm extends the lifetime of the network for a few more rounds.

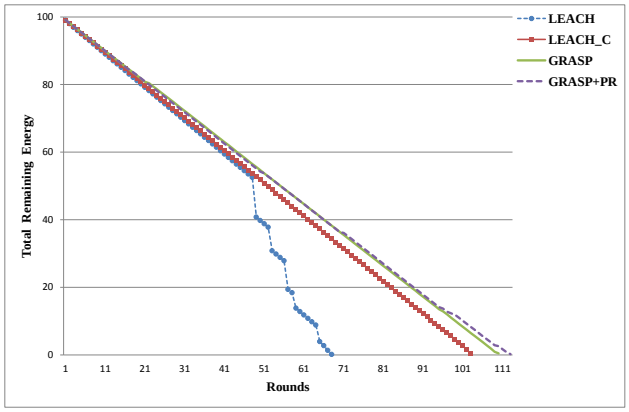


Fig. 11 – Total Energy Consumption in Each Round

Figures 12 and 13 show the dissipated energy with respect to the number of transmissions, for two instances with 100 and 200 sensors, respectively. With LEACH method, after 50% of the dissipated energy, there is a considerable decreasing in the amount of transmissions. For the other algorithms, the energy consumption with respect to the number of transmissions has a more homogeneous behavior. The GRASP+PR algorithm performs more transmissions than the other methods.

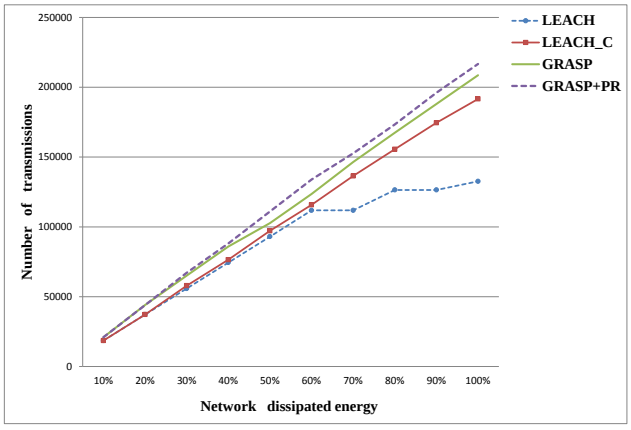


Fig. 12 – Energy dissipation for an instance of 100 sensors

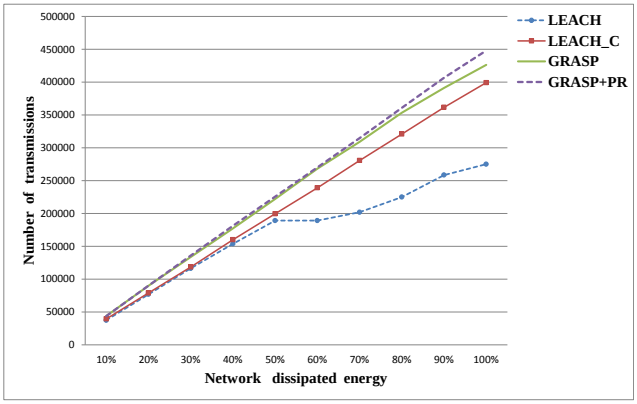


Fig. 13 – Energy dissipation for an instance of 200 sensors

Table 3 presents the average number of successful transmissions performed for each algorithm, for each set of instances. We can see that the GRASP algorithms perform considerably better than the LEACH and LEACH-C methods. GRASP+PR slightly outperforms the basic version GRASP. Figure 14 shows the number of transmissions (lifetime) for each instance. We can note that the number of transmissions increases by according to the amount of sensors.

Table 3 – Average of number of transmissions

| *n* | LEACH | LEACH_C | GRASP | GRASP+PR |
|---|---|---|---|---|
| 50 | 65231.5 | 91640.4 | 87275.5 | 90126.4 |
| 60 | 79632.6 | 109528.0 | 112714.7 | 118008.4 |
| 70 | 90611.0 | 131236.5 | 127127.8 | 138851.4 |
| 80 | 108139.2 | 150028.2 | 160466.1 | 167897.3 |
| 90 | 118401.2 | 173182.6 | 184680.2 | 185643.0 |
| 100 | 138264 | 191761.6 | 205966.9 | 214793.8 |
| 125 | 175544.1 | 244124.9 | 267060.8 | 273883.8 |
| 150 | 211103.5 | 296543.0 | 325567.5 | 330813.5 |
| 200 | 288759.9 | 398746.1 | 436972.9 | 449902.2 |
| 300 | 441693.3 | 608445.9 | 671031.0 | 683577.1 |
| **Average** | **171738.0** | **239523.7** | **257886.3** | **265349.7** |

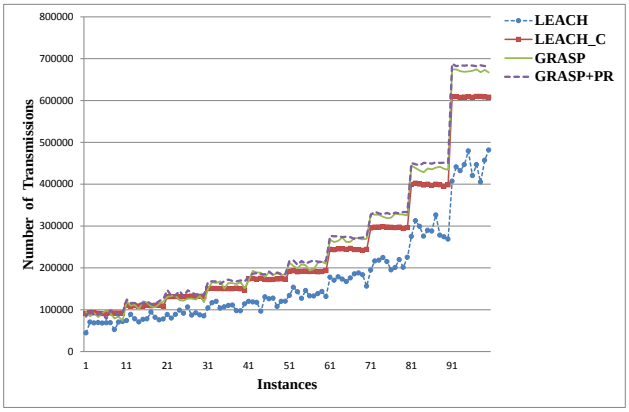


Fig. 14 – Total number of transmissions for each instance

The impact of the network lifetime is analyzed by the number of rounds that the network remains alive (network longevity). In Figure 15 the algorithms are compared, over all the instances, by Average RPD (ARPD) related to the number of rounds that the network remains alive. The RPD is computed by the following Equation:

$$RPD\% = 100 \times \frac{nR_{best} - nR_{algorithm}}{nR_{best}} \quad (6)$$

where, $nR_{algorithm}$ is the number of rounds determined by an algorithm and $nR_{best}$ is the largest number of rounds obtained among all the algorithms. We can see that the algorithms GRASP and GRASP+PR extend the network lifetimes when comparing with LEACH and LEACH-C. In the most of the instances, on average, GRASP+PR performs better than the simple GRASP algorithm.

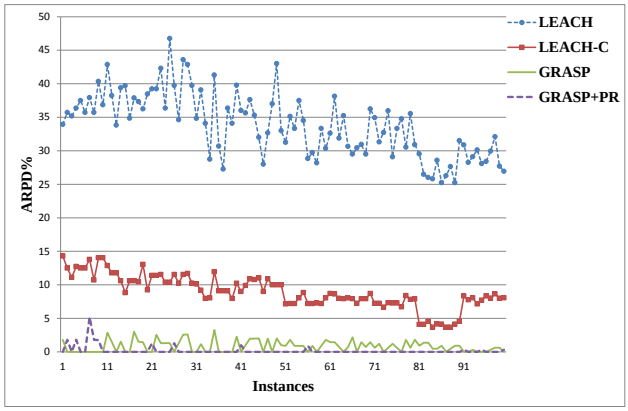


Fig. 15 – Network Longevity

## 6. Conclusion

In this work, we have proposed a GRASP based algorithm for the cluster formation problem in wireless sensor networks. An intensification phase based on the Path Relinking technique was used to improve the quality of the obtained solutions. We conducted simulation experiments to measure the performance of our clustering algorithms GRASP and GRASP+PR. Performance evaluation showed that GRASP+PR provides a better energy savings and better network lifetime than both LEACH and LEACH-C protocols. In conclusion, our presented method can now be considered state-of-the-art heuristics for the cluster formation problem in in wireless sensors networks.

As future work, we intend to extend our approach to form hierarchical multihop clusters. Due to the sensor limitation, a more realistic simulation can be done by considering the maximum range transmission of sensors.

**Acknowledgments**

This research was supported by CNPq and FAPEMIG.

## References

[1] A. A. Abbasi and M. Younis, ”A survey on clustering algorithms for wireless sensor networks”. Computer Communications 30, 2007, pp. 2826-2841.

[2] N. M. Abdul Latiff et al. ”Energy-aware clustering for wireless sensor networks using Particle Swarm Optimization”. The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2007.

[3] P. Agarwal and C. Procopiuc. ”Exact and approximation algorithms for clustering”. Proc. 9th Annu. ACM-SIAM Symp. Discrete Algorithms, Baltimore, MD, Jan. 1999, pp. 658-667.

[4] I.F. Akyildiz et al. ”Wireless sensor networks: a survey”. Computer Networks 38, 2002, pp. 393-422.

[5] T. A. Feo and M. G. C. Resende,”Greedy randomized adaptive search procedures”. Journal of Global Optimization 6, 1995, pp. 109-133.

[6] F. Glover. ”Tabu search and adaptive memory programing”. Interface in Computer Science and Operational Reaserch, 1996, pp. 1-75.

[7] W.B. Heinzelman et al.”An application specific protocol architecture for wireless microsensor networks”. IEEE Transactions on Wireless Networking, vol. 1, no.4, 2002, pp. 660-670.

[8] W.B. Heinzelman et al.”Energy-Efficient Communication Protocol for Wireless Microsensor Networks”. 33rd Hawaii International Conference on System Sciences, 2000.

[9] S. Jin et al. ”Sensor network optimization using a genetic algorithm”. Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics, 2003.

[10] S. Lindsey et al. ”Data gathering algorithms in sensor networks using energy metrics”, IEEE Transactions on Parallel and Distributed System, vol. 13, no. 9, 2002, pp. 924-935.

[11] D. Montegomery. ”Design and Analysis of Experiments”. John Willy and Sons, 2007.

[12] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, A centralized energy-efficient routing protocol for wireless sensor networks, IEEE Radio Communications, pp. S8-S13, 2005.

[13] R Rajagopalan and P. K. Varshney, ”Data aggregation techniques in sensor networks: A survey”. IEEE Communication Surveys and Tutorials, Vol. 8, No. 4, December 2006, pp.48-63.

[14] T. Rappaport, ”Wireless Communications: Principles and Practice”. Prentice-Hall, Englewood Cliffs, NJ, 1996.

[15] M. G. C. Resende and C. C. Ribeiro. ”Metaheuristics: Progress as real problem solvers”. Kluwer Academic Publishers, 2005.

[16] M. G. C. Resende and R. F. Werneck, ”A Hybrid Heuristic for the p-Median Problem”. Technical Report TD-5NWRCR, AT&T Labs Research, 2003.

[17] A. Salehpour et al. ”Efficient Clustering of Wireless Sensor Networks Based on Memetic Algorithm” [ed.] University of Tehran, 2008.

[18] J. Tillet et all,”Cluster-head identification in ad hoc sensor networks using particle swarm optimization”. IEEE International Conference on Personal Wireless Communications, 2002, pp. 201-205.

[19] S. Yi et all. ”PEACH: power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks”. Computer Communications, vol. 30, 2007, pp. 2842-2852.