A Hybrid Approach to solve the Coverage and Connectivity Problem in Wireless Sensor Networks

Frederico Paiva Quintão[1], Fabíola Guerra Nakamura[1] and Geraldo Robson Mateus[1]

[1]Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, MG – Brazil

{fred,fgnaka,mateus}@dcc.ufmg.br

Abstract. *The recent area of Wireless Sensor Networks (WSNs) has brought new challenges to developers of network protocols. One of these challenges consists of maintaining the coverage of the monitoring area and the connectivity between the network nodes, preferably accomplishing the management of the network resources. This problem can be modeled as a Mathematical Programming problem, but it requires a hard computational effort and, since the WSNs may be very dynamic, any slow management decision can lead to serious problems. In this work, the problem was decomposed into two sub-problems and solved by a hybrid approach, which consists of two phases: a genetic algorithm and a local search based on two classical graph algorithms.*

1. Introduction

A Wireless Sensor Network (WSN) is a new kind of ad-hoc network, with distributed sensing and processing capacities. WSNs can be composed of tens to hundreds of small battery-powered devices, called sensor nodes, and they can be used in a large number of applications, such as indoor environments control, air pollution level monitoration, in assembly lines and as military spies, providing information about enemies movements.

Figure 1(a) shows a common WSN architecture: many sensor nodes monitoring an area and reporting information to the sink node, which is a special node used to send information outside the network. Figure 1(b) shows two examples of sensor nodes: Mica2 (the biggest one) and Mica2Dot [Crossbow Technology, 2003], which can perform activities like sensing, communication and processing. Usually the hardware of a sensor node includes:

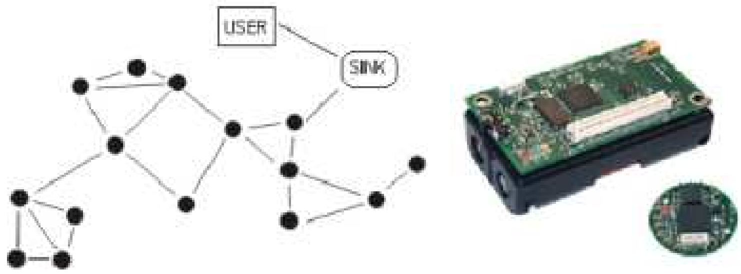
- A sensor board, with at least a kind of sensor on it;
- A limited quantity of memory (128 KB in the case of Mica Motes);
- A processor, with limited power of processing (8 MHz for Mica Motes);
- A radio to perform wireless communication, generally under IEEE 802.11 specifications;
- A battery, which provides energy to all the other components.

Since a WSN can be deployed into a hostile area (such as a volcano crater), and its number of nodes can be high, recharging or replacing nodes' battery may be inconvenient. So, the development of power-saving protocols for the organization of these networks can extend their lifetime, what is very desirable. Thus, in this paper we propose an approach that performs this organization, from the WSN's management point of view. Our main goals are:

- To turn the minimal number of nodes on, controlling the density and indirectly reducing some problems like radio interference, collision of packets and media congestion [Tilak et al., 2002];
- To ensure that this number of active nodes can cover the monitoring area;
- To guarantee that the information can flow outside, that is, the active nodes are connected to the sink;
- To extend the network lifetime.

This problem is known as CCP-WSN (Coverage and Connectivity Problem in Wireless Sensor Networks). CCP-WSN is a network-design-like problem and can be modeled as a mixed integer linear programming (MILP) problem, but as we will be seen, its formulation requires a hard computational effort. To achieve good solutions in a feasible processing time, the problem was decomposed into two sub-problems, which are solved by a hybrid approach, which works in a GRASP-fashion way: first it uses a genetic algorithm for a stochastic search, and then it applies a local search, based on Prim [Ziviani, 2003] and Dijkstra [Ziviani, 2003] [Tanembaum, 1996] algorithms.

The remainder of the paper is organized as follows: in section 2 we detail some related works; in section 3 we introduce CCP-WSN mathematical formulation; in section 4 we describe CCP-WSN decomposition and present the two sub-problems, as well the algorithms proposed to solve them; some computational results and comparisons between the exact model and our hybrid approach are reported in section 5. Finally, we present our conclusions and future work in section 5.



(a) A common architecture of a WSN.

(b) Sensor nodes of the family Mica Motes.

Figure 1: Wireless Sensor Networks.

2. Related work

Some papers have proposed protocols and algorithms to extend the lifetime of a WSN, through an intelligent management of its resources, but the most discuss only the Coverage Problem. Vieira *et al* [Vieira et al., 2003] use a technique based on the Voronoi algorithm to discover backup nodes (redundant nodes which can be turned off to safe energy). The Voronoi algorithm is also used by Meguerdichian *et al* [Meguerdichian et al., 2001], to discover dense and non-dense areas of the network.

An integer linear programming formulation for the Coverage Problem is proposed by Megerian and Potkonjak [Megerian and Potkonjak, 2003]. The Coverage Problem is modeled using the Set Covering Problem in [Slijepcevic and Potkonjak, 2001], and a heuristic is presented to solve it. The modeling based on the Set Covering Problem is also used by [Siqueira et al., 2003], which also takes some connectivity assumptions to keep the whole network connected.

Meta-heuristics have already been used in the field of WSNs. Heinzelman *et al* [Heinzelman et al., 2002] present LEACH-C, a routing protocol for hierarchical WSNs. A Simulated Annealing algorithm is used to compute the best set of cluster-heads (group leaders) nodes. Quintão *et al* in [Quintao et al., 2004] propose a genetic algorithm for the *on-demand* Coverage Problem.

In [Nakamura, 2003] and [Menezes, 2004] there are some formulations based on Mixed Integer Linear Programming (MILP), which solve the Coverage and Connectivity Problem in WSNs. The work of Nakamura presents a dynamic node scheduling for all the network lifetime. Menezes solves the problem using Lagrangean Relaxation, and also considers scenarios with obstacles.

Problems generated by high density of nodes are discussed by Tilak *et al* [Tilak et al., 2002]. A management architecture for WSNs (as well its impact over the network working) is presented by Ruiz *et al* [Ruiz, 2003].

3. CCP-WSN Mathematical Formulation

Our problem can be stated as: *Given a monitoring area A, a set of demand points D, a set of sensor nodes S and a sink node m, the Coverage and Connectivity Problem in Wireless Sensor Networks (CCP-WSN) consists of assuring that at least n sensor nodes from S are covering each demand point $j \in D$ in the monitoring area A, and that there is a path between these nodes and the sink node m.* CCP-WSN is formulated as a mixed integer linear programming (MILP) problem. The following parameters are used in our formulation:

S set of sensor nodes
D set of demand points
A^d set of arcs connecting sensor nodes to demand points
A^s set of arcs connecting sensor nodes
A^m set of arcs connecting sensor nodes to the sink node
$I^d(A)$ set of arcs $(i, j) \in A^d$ incoming on the demand point $j \in D$
$I^s(A)$ set of arcs $(i, j) \in A^s \cup A^d$ incoming on the sensor node $j \in S$
$O^s(A)$ set of arcs $(i, j) \in A^s \cup A^m$ outgoing the sensor node $i \in S$
n coverage precision that defines how many nodes should cover a demand point
ME node maintenance energy
TE node transmission energy
RE node reception energy
NC coverage penalty, cost of no coverage of a demand point

The model variables are:

x_{ij} variable that has value 1 if node i covers demand point j, and 0 otherwise
z_{lij} decision variable that has value 1 if arc (i,j) is in the path between sensor node l and the sink node m, and 0 otherwise
y_i decision variable that has value 1 if node i is active, and 0 otherwise
h_j variable to indicate if demand point j is not covered
e_i variable to indicate the energy consumed by node i

The formulation proposed is presented below.

$$min \sum_{i \in S} e_i + \sum_{j \in D} NC_j \times h_j \tag{1}$$

The objective function (1) minimizes the network energy consumption and the number of not covered demand points. Since we minimize the network energy consumption, the mathematical formulation is indirectly reducing the number of active nodes, regarding to equation (10). Of course our objective function does not correspond to all the goals stated in Section 1 (Introduction), but the result of our whole model deals with them. According to the work presented in [Vieira et al., 2003] and the results from [Tilak et al., 2002], it was suggested that the density control in a WSN can reduce problems like radio interference between neighbors nodes, as well collision of packets and media congestion.

The constraints (2), (3), (4), and (5) deal with the coverage problem. They assure that the active nodes cover the demand points (we always consider a demand point per square meter).

$$\sum_{ij \in I_j^d(A^d)} x_{ij} + h_j \geq n, \forall j \in D \tag{2}$$

$$x_{ij} \leq y_i, \forall i \in S, \forall ij \in A^d \tag{3}$$

$$0 \leq x_{ij} \leq 1, \forall ij \in A^d \tag{4}$$

$$h_j \geq 0, \forall j \in D \tag{5}$$

The constraints (6), (7), (8) and (9) are related to the connectivity problem. They assure a path between each active sensor node $l \in S$ and the sink node m.

$$\sum_{ij \in I_j^s(A^s)} z_{lij} - \sum_{jk \in O_j^s(A^s \cup A^m)} z_{ljk} = 0, \forall j \in (S \cup m - l), \forall l \in S \tag{6}$$

$$- \sum_{jk \in O_j^s(A^s \cup A^m)} z_{ljk} = -y_l, j = l, \forall l \in S \tag{7}$$

$$z_{lij} \leq y_i, \forall i \in S, \forall l \in (S - j), \forall ij \in (A^s \cup A^m) \tag{8}$$

$$z_{lij} \leq y_j, \forall j \in S, \forall l \in (S - j), \forall ij \in (A^s \cup A^m) \tag{9}$$

The energy constraints (10) and (11) define the energy limit values. Constraints (11) also justify why we declared our model as a Mixed Integer Linear Programming problem (variable e_i can have real values).

$$\begin{aligned} &ME_i \times y_i + \sum_{l \in (S-i)} \sum_{ki \in I_i^s(A^s \cup A^m)} RE_i \times z_{lki} + \\ &\sum_{l \in S} \sum_{ij \in O_i^s(A^s \cup A^m)} TE_{ij} \times z_{lij} \leq e_i, \forall i \in S \end{aligned} \tag{10}$$

Note that a node spends its energy with self-maintenance and with transmission and reception of packets.

$$e_i \geq 0, \forall i \in S \tag{11}$$

The constraints (12) define the decision variables as boolean, and the constraints (13) define the others variables as real.

$$y, z \in \{0, 1\} \tag{12}$$

$$x, h, e \in \Re \tag{13}$$

The model solution consists of a subset of active nodes and also reports which demand points are not covered, assuring the best possible coverage, and provides a path between the active nodes and the sink node, assuring the network connectivity. The solution also estimates the network energy consumption.

4. CCP-WSN decomposition

How CCP-WSN requires hard computational effort for optimal solutions, it was decomposed into two sub-problems; the used strategy consists of:

1. First a Coverage Problem is solved, finding the minimal number of nodes needed to cover all the monitoring area; for this sub-problem it is used a genetic algorithm;
2. A local search in the best solution found in the previous step is made to ensure the connectivity between the active nodes. In this step Prim's and Dijkstra's algorithms are applied.

A heuristic based on genetic search was chosen to solve the Coverage Problem because this kind of algorithm usually provides more than one good (feasible) solution, and maybe this redundance can be interesting to the manager of the network. Otherwise, we do not try to solve all the problem with a genetic algorithm because it is difficult to keep the feasibility of the solutions in the original CCP-WSN, since it is a problem with many constraints. We would have to search for very special operators, to avoid the generation of non-feasible solutions, what could involve even hard computational tasks.

4.1. Coverage Problem Formulation and proposed genetic algorithm

We use the Coverage Problem formulation proposed by Nakamura [Nakamura, 2003]. The problem can be stated as: *Given a monitoring area* A*, a set of sensor nodes* S *and a set of demand points* D*, the Coverage Problem consists of assuring that at least one sensor node* $s \in S$ *will cover each demand point* $j \in D$. The formulation uses the previous parameters and the energy cost AE to turn a sensor on.

The model can be formulated as:

$$min \sum_{i \in S} AE_i \times y_i + \sum_{j \in D} NC_j \times h_j \tag{14}$$

subject to:

$$\sum_{ij} x_{ij} + h_j \geq 1, \forall j \in D \text{ and } \forall ij \in A^d \tag{15}$$

$$x_{ij} \leq y_i, \forall i \in S \text{ and } \forall ij \in A^d \tag{16}$$

$$0 \leq x_{ij} \leq 1, \forall ij \in A^d \tag{17}$$

$$h_j \geq 0, \forall j \in D \tag{18}$$

$$y \in \{0, 1\} \tag{19}$$

$$x, h \in \Re \tag{20}$$

The objective function minimizes the number of active nodes and the number of uncovered demand points. Constraints (15) assure that each demand point may be covered by a sensor node or keep uncovered. Constraints (16) impose that a node only can sense if it is active. The remainder constraints deal about variables' limits.

In order to improve our whole algorithm efficiency, we changed the Coverage Problem objective function as follows:

$$min \sum_{i \in S} (AE_i + PC_i) \times y_i + \sum_{j \in D} NC_j \times h_j \tag{21}$$

where PC_i is a variable that contains the cost of the path from each node $i \in S$ to the sink node m (computed by the Dijkstra's algorithm applied to all nodes of the network, during a pre-processing phase). The variable PC_i is used as a penalty for those nodes whose path to the sink node is expensive, and the results show that this change is a bit interesting, improving our algorithm (in fact, this is a way for looking to the Connectivity Problem during the genetic algorithm search).

To solve the changed Coverage Problem we propose a genetic algorithm based on binary encoding, as described below:

4.1.1. Encoding

We use binary encoding of parameters. Each chromosome has size T equal to the number of sensor nodes in the network. Each position of the chromosome represents a gene: if a chromosome position is set to 1, this implies that the node corresponding to this position is turned on in this chromosome. For example, suppose a network containing 10 nodes. Also suppose that one of the chromosomes has the following nodes activated: (0, 2, 7, 9); so, its binary representation would be as follows:

1	0	1	0	0	0	0	1	0	1

Given a set of active nodes from S, we can evaluate the Coverage through a binary coverage matrix, which reports, for an input (i, d), if sensor node i covers demand point d.

4.1.2. Genetic algorithm general specification

In our algorithm the following operators are implemented:

- Selection: uses Cost Weighting Pairing [Haupt and Haupt, 1998]. When two chromosomes are selected, they are always combined;
- Recombination (crossing over): implements the easiest way, in which a random number c is generated and in this position the crossing over takes place. Each couple of recombined chromosomes generates two offsprings, which go to the place of the worst chromosomes;
- Mutation: happens with a small probability μ. The used value (as well as the population size and number of generations) is described in the *Computational results* section.

4.2. Ensuring connectivity to CCP-WSN

The solution generated by the genetic algorithm random search may cover well the monitoring area, but maybe some nodes can be disconnected. This is a problem because the information could not flow outside and arrive to the final user. So we need a strategy to turn the network into a connected network. For this task Prim's and Dijkstra's algorithms are used. Our strategy consists of two phases:

1. Initially we apply Prim's Minimum Spanning Tree (MST) algorithm over a graph G_1 containing the active nodes of the network (set during the last step). The condition for an edge belong to G_1 is the following: *An edge (u, v) can belong to G_1 only if the distance between nodes u and $v \in G_1$ is shorter than the max communication range of the nodes*. The result of the application of this MST algorithm is a tree that could be used as a routing tree. Therefore, given the condition above, some of the active nodes may be disconnected from the tree. So, we apply the next step.
2. A graph G_2 is created, containing all the nodes of the network. The same condition described above is applied: *An edge (u, v) can belong to G_2 only if the distance between nodes u and $v \in G_2$ is shorter than the max communication range of the nodes*. So, the Dijkstra's shortest path algorithm is applied from each one of the disconnected nodes to the sink node. Actually, this path may go across non-active nodes (otherwise all active nodes would be connected during the last step). Thus, we also turn these not active nodes on.

4.3. Final algorithm

Algorithm 1 summarizes the hybrid approach used to solve the CCP-WSN.

Algorithm 1 Hybrid(set S of sensor nodes)

```
for all i ∈ S do
    PC_i ← Compute Dijkstra(i, sink);
end for
Compute Coverage Matrix; /*notices if a sensor node i covers a demand point j*/
Connectivity Matrix; /*notices the Euclidian distance between a sensor node i and another node l*/
/*Genetic algorithm*/
Create random initial population, with size |PopInitial|;
Evaluate initial population using equation (21);
ShellSort (Initial population); /*Sort*/
Natural Selection(Initial Population); /*removes the worst chromosomes from the initial population*/
while NOT stop condition do
    Select chromosomes for mating using Cost Weighting Pairing;
    Proceed matching/crossing over;
    Mutation with probability μ;
    Evaluate new population using equation (21);
    Shellsort(new Population);
end while
/*end of genetic algorithm*/

/*Local search for connectivity ensuring*/
Compute Prim(active nodes);
while there is a not connected node i do
    Compute Dijkstra(i, sink);
end while
/*end of local search*/
Compute objective function (1);
/*end of hybrid algorithm*/
```

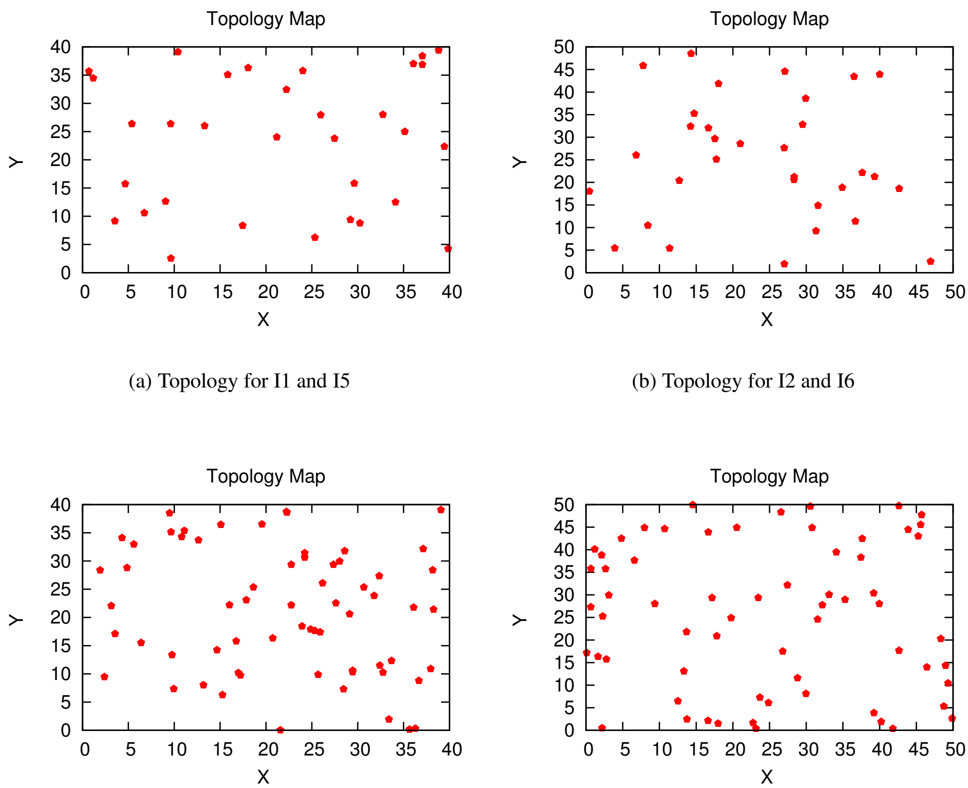
5. Computational results

In this section we report some of our results that were obtained using our CCP-WSN implementation in CPLEX 7.0 and our hybrid algorithm. First of all we define some parameters for our genetic algorithm (some of them were also used in CPLEX). Most of the values are obtained from the analysis of the results from [Nakamura, 2003] [1] and are showed in Table 1.

Parameter	Value	Description
ME	18	Maintenance energy
AE	18	Cost to turn a sensor node on
RE	2	Reception energy
NC	10000	Coverage penalty
$PopInitial$	1400	Number of chromosomes in the initial population
$SizePop$	600	Number of chromosomes in each generation
$Sensing\_range$	15	Sensing range of all sensor nodes
μ	10%	Mutation probability
$MaxGenerations$	25	Max number of generations (genetic algorithm stop condition)
T_{tx}	0.25	Time during a sensor node will transmit data

Table 1: Parameters values.

Eight instances named I1, I2, I3, I4, I5, I6, I7 and I8 are used for our tests. We consider square monitoring areas and the sink node positioned in the center of the area. Instance I1 consists of 32 nodes in a square area of $40m \times 40m$. I2 consists of 32 nodes in an area of $50m \times 50m$, I3 contains 64 nodes in an area of $40m \times 40m$, and I4 consists of 64 nodes in an area of $50m \times 50m$. For these instances we considered the max communication range of $20m$. Instances I5, I6, I7 and I8 contain the same topology configuration of I1, I2, I3 and I4, but for these instances the max communication range of $25m$ is considered. Figure 2 illustrates the topology maps used in our simulations. We run the CCP-WSN on the commercial software CPLEX 7.0 and our algorithm for the 8 instances



(a) Topology for I1 and I5

(b) Topology for I2 and I6

(c) Topology for I3 and I7

(d) Topology for I4 and I8

Figure 2: Topology maps used for computational analysis (generated randomly).

described above. As the genetic algorithm performs a stochastic search, we run each instance 33 times and take

[1] transmission energy (TE) value was obtained from MicaMotes Manual [Crossbow Technology, 2003] and changes in accordance with distance.

medium values and standard deviation (represented as $\Delta()$ in the tables). Table 2 shows the results obtained by CPLEX and Table 3 presents the results obtained by our algorithm:

Instance	Active nodes	Coverage	Time (sec)	Objective
I1	5	100%	347.2	101.525
I2	9	100%	239.83	200.025
I3	5	100%	1816.26	101.350
I4	8	100%	21408.67	179.600
I5	5	100%	809.30	101.525
I6	9	100%	353.33	192.175
I7	5	100%	5106.56	101.400
I8	7	100%	5575.84	147.525

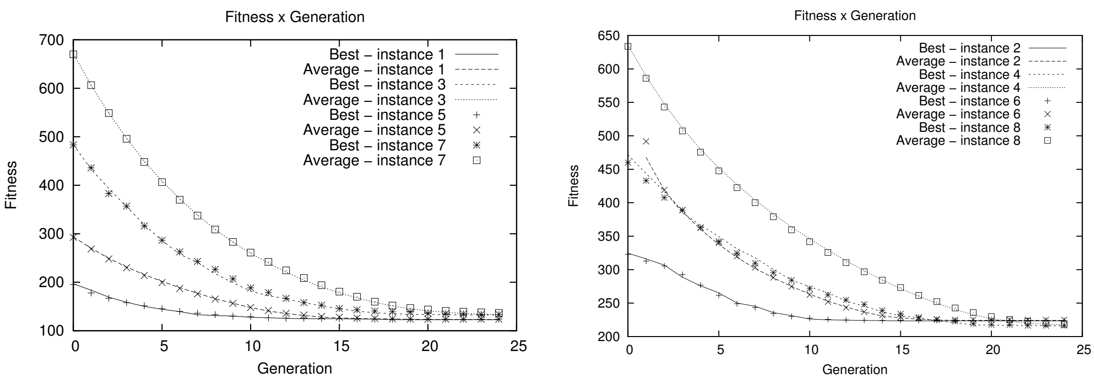
Table 2: CPLEX optimal solutions.

Instance	Active nodes	Δ(Active)	Time (sec)	Δ(Time)	Objective	Δ(Objective)
I1	5.303	0.054	51.849	1.953	117.279	12.321
I2	11.090	0.196	92.061	0.527	273.580	34.923
I3	5.333	0.060	97.601	0.804	116.929	24.449
I4	10.393	0.073	163.920	1.118	250.505	28.046
I5	5.363	0.113	51.499	0.496	118.456	14.6821
I6	9.000	0.000	91.890	0.606	227.145	10.365
I7	5.090	0.016	99.894	13.347	108.478	7.728
I8	9.151	0.027	192.932	20.402	226.749	26.515

Table 3: Hybrid algorithm results.

The hybrid algorithm always reaches 100% of coverage of the monitoring area in all tests for all instances, and the results show that the whole algorithm gets on in all instances, regarding the number of active nodes, and solves fastly all of them, including the largest ones (what is a quite interesting since the WSNs can be very dynamic and the manager should answer fastly to the changes). The value of the objective function is also close to the optimal for the smallest instances. The run time of the pos-processing phase was also computed. We discovered this phase spends no significant time. It happens because the algorithms used (Prim and Dijkstra) are both very efficient: for a graph $G(V, E)$, where V is the set of vertices and E is the set of edges, Prim's algorithms runs on $O(E \log V)$ and our implementation of Dijkstra algorithm [Tanembaum, 1996] is polynomial.

Figure 3 illustrates the medium evolution of the genetic algorithm used to solve the changed Coverage Problem for each group of instances. We notice the algorithm convergence generally happens with soft curves, what is interesting because it can avoid local minimum. As expected, for some instances the behavior of the algorithm is similar, for example, I1 & I5, I3 & I7 and I2 & I6, since they have the same topology and the genetic algorithm just deals to the *changed* Coverage Problem (in fact overlapping can be detected on the graphs). This observation shows that the shortest paths found when we are using a communication range of $25m$ are not much better than those when the communication range is of $20m$ (at least in our instances, of course this result can not be true for all cases and for different sizes of ranges).



(a) Fitness for Instances I1, I3, I5 and I7

(b) Fitness for Instances I2, I4, I6 and I8

Figure 3: Genetic algorithm evolution for all instances.

To obtain more fonts of comparison, we tried to get results with CPLEX running in the same amount of time of our algorithm, but no feasible solution was found. These results confirm the advantage of the use of the hybrid algorithm in those situations where the processing time plays an important role.

6. Conclusions

In this paper we propose a hybrid approach to solve the Coverage and Connectivity Problem in Wireless Sensor Networks. Our algorithm could run together with a WSN management architecture, like the one proposed in [Ruiz, 2003]. Our results show that our algorithm performs well in some scenarios, and runs fast over all of them, including the largest ones. This is a important observation since WSNs may be very dynamic, and the manager should react fastly to the changes, otherwise serious problems such death of nodes or loses in the degree of QoS (Quality of Service) can occur. Simulations have been developed in the software Network Simulator (ns-2) to validate our results.

As future work we intend to compute results for new instances and to develop new strategies to solve the Connectivity Problem, dealing more with it during the genetic algorithm phase (in fact, we are trying to discover an encoding and soft operations suitable for our whole problem). We also intend to develop some strategies to improve the genetic algorithm, using hybrid operations, like GRASP's path-relinking during the chromosomes matching. We also would like to use our theorical results as a background for the development of distributed algorithms for topology control of WSN, what would be very interesting given the ad-hoc characteristics of these nets.

Acknowledgments

This work has been supported in part by CNPq/Brazil (process 18.2709/2003-5). We are grateful to Felipe de Castro Louback Rocha (Federal University of Minas Gerais) and the anonymous reviewers for their valuable comments on this work.

References

Crossbow Technology, I. (2003). *MPR - Mote Processor Radio Board MIB - Mote Interface / Programming Board User Manual*. Crossbow Technology, Inc.

Haupt, R. L. and Haupt, S. E. (1998). *Practical Genetic Algorithms*. John Wiley & Sons, Inc.

Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. (2002). An Application-Specific Protocol Architeture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*.

Megerian, S. and Potkonjak, M. (2003). Low Power 0/1 Coverage and Scheduling Techniques in Sensor Networks. UCLA. Technical report.

Meguerdichian, S., Koushanfar, F., Potkonjak, M., and Srivastava, M. B. (2001). Coverage Problems in Wireless ad-hoc Sensor Networks. In *INFOCOM' 01*. IEEE.

Menezes, G. C. (2004). Modelos e Algoritmos para definição da Densidade e Posicionamento dos nós em uma Rede de Sensores sem fio. Master's thesis, Universidade Federal de Minas Gerais (in portuguese).

Nakamura, F. G. (2003). Planejamento Dinâmico para Controle de Cobertura e Conectividade em Redes de Sensores sem fio planas. Master's thesis, Universidade Federal de Minas Gerais (in portuguese).

Quintao, F. P., Mateus, G. R., and Nakamura, F. G. (2004). An Evolutive Approach for the Coverage Problem in Wireless Sensor Networks. In *Proceeding of 24th Brazilian Computer Society Congress - 1st Place at the 23rd Undergraduate Scientific Research Contest*. Brazilian Computer Society (in portuguese).

Ruiz, L. B. (2003). *Manna: A Management Architecture for Wireless Sensor Networks*. PhD thesis, Universidade Federal de Minas Gerais.

Siqueira, I., Ruiz, L., Loureiro, A., and Nogueira, J. M. (2003). A Management Service for Density Control in Wireless Sensor Networks. In *22nd Brazilian Simposium on Computer Networks (SBRC)*. Brazilian Computer Society (in portuguese).

Slijepcevic, S. and Potkonjak, M. (2001). Power Efficient Organization of Wireless Sensor Networks. In *IEEE International Conference on Communications (ICC) 2001*. IEEE.

Tanembaum, A. S. (1996). *Computer Networks, 3rd Edition*. Prentice Hall PTR.

Tilak, S., Abu-Ghazaleh, N., and Heinzelman, W. (2002). Infrastructure Tradeoffs for Sensor Networks. In *ACM 1st International Workshop on Sensor Networks and Applications (WSNA'02)*. ACM.

Vieira, M. A. M., Vieira, L. F. M., iz, L. B. R., Loureiro, A. A. F., Fernandes, A. O., and Nogueira, J. M. S. (2003). Scheduling Nodes in Wireless Sensor Networks: A Voronoi Approach. In *28th Annual IEEE International Conference on Local Computer Netwo rks*. IEEE.

Ziviani, N. (2003). *Projetos de Algoritmos, com Implementações em Pascal e C*. Thompson Editora.