Journal of Universal Computer Science, vol. 14, no. 15 (2008), 2549-2565
submitted: 13/3/08, accepted: 30/6/08, appeared: 1/8/08 © J.UCS

Optimal Sensor Network Layout Using Multi-Objective Metaheuristics

Guillermo Molina
(University of Málaga, Spain
guillermo@lcc.uma.es)

Enrique Alba
(University of Málaga, Spain
eat@lcc.uma.es)

El-Ghazali Talbi
(University of Lille, France
El-ghazali.Talbi@lifl.fr)

Abstract: Wireless Sensor Networks (WSN) allow, thanks to the use of small wireless devices known as sensor nodes, the monitorization of wide and remote areas with precision and liveness unseen to the date without the intervention of a human operator. For many WSN applications it is fundamental to achieve full coverage of the terrain monitored, known as *sensor field*. The next major concerns are the energetic efficiency of the network, in order to increase its lifetime, and having the minimum possible number of sensor nodes, in order to reduce the network cost. The task of placing the sensor nodes while addressing these objectives is known as WSN layout problem. In this paper we address a WSN layout problem instance in which full coverage is treated as a constraint while the other two objectives are optimized using a multi-objective approach. We employ a set of multi-objective optimization algorithms for this problem where we define the energy efficiency and the number of nodes as the independent optimization objectives. Our results prove the efficiency of multi-objective metaheuristics to solve this kind of problem and encourage further research on more realistic instances and more constrained scenarios.

Key Words: sensor networks, multiobjective optimization, metaheuristics
Category: G.1.6, I.2.8.

1 Introduction

Wireless Sensor Networks (WSN) have recently become one of the hot topics in research [Akyildiz et al. 2002, Culler et al. 2004]. Their capabilities for monitoring large areas, accessing remote places, real-time reacting, and relative ease of use have brought scientists a whole new horizon of possibilities. WSN have so far been employed in militar activities such as reconnaissance, surveillance [Lédecze et al. 2005] and target acquisition [Nemeroff et al. 2001], environmental activities such as forest fire prevention [Mladineo and Knezic 2000], geophysical activities such as volcano activity study [Werner-Allen et al. 2006], biomedical purposes such as health data monitoring [Yuce et al. 2007] or artificial retina

[Schwiebert et al. 2001], or civil engineering such as structural health measurement [Xu et al. 2004]. Their uses increase by the day and their potential applications seem boundless. The wide variety of applications results in a wide variety of networks bearing different constraints and having different features, yet most of them share some common issues that allow them to be treated homogenously.

When deploying a WSN the positioning of the sensor nodes becomes one of the major concerns. For some applications the coverage of the network, which depends directly on the positions of the nodes, is a critical issue. In the countersniper system [Lédecze et al. 2005] a sensor network is placed to secure an area from possible snipers. The network is designed to be placed in an urban scenario, where snipers can easily hide, and the nodes are equipped with sound sensors capable of detecting the blast produced by the bullet. By retrieving this information from several nodes the WSN is able to perform a space-temporal data process in such a way that the position of the shooter can be estimated. In such a critical system where lifes are on the line it is absolutely essential that the sensors can detect the bullet trajectory no matter what this trajectory is; therefore complete terrain coverage is a must. There are other applications for which these concerns do not apply; for instance in a biomedical sensor network the coverage is not an issue (since it is easily achieved), but the interface with the patient and the safety of the system become of a critical importance.

There are two other important concerns. In the first place a WSN that is likely to be deployed in a battleground, where it is unfeasible to provide it neither support nor maintenance, has to operate by itself for the maximum possible time. This causes energy saving to be one of the principal policies in a WSN in order to increase the network lifetime. In the second place the WSN has to operate as stealthy as possible so as to not be detected by the enemy. This implies that the number of sensors has to be kept at a minimum in order to reduce the probability of one of them being discovered. Besides, these concerns are no strangers to civilian-purpose sensor networks, since either increasing the lifetime or reducing the number of sensors will affect in a cost reduction of the network.

Since many WSN have large numbers of nodes and some of their features cannot be calculated in a simple manner, the task of selecting the geographical positions of the nodes for an optimal resulting network -referred to as WNS layout problem- can be very complex. Besides, this problem recalls the unicost set covering problem, which is known to be NP. Therefore, metaheuristics seem an interesting option to solve this problem. In this paper we propose a solution method for the WSN layout problem using the Paradiseo framework [Liefooghe et al. 2007]. In our problem definition we focus on minimizing both the energy depletion of the nodes in the network and the number of nodes while the coverage obtained by the network is considered as a constraint (full coverage of the *sensor field* required).

When a WSN has to be displayed there are many considerations that have to be dealt with. The coverage has to fulfill some restrictions, and it is likely that the highest coverage possible will be preferred. The number of sensor nodes, while in principle is not as restrictive as the first, should also be kept low for economy reasons. Finally the energy management of the network is a key issue that has to be taken into careful consideration since the lifetime of the network depends on it.

The rest of the paper is organized as follows. The existing state of the art is briefly reviewed in Section 2. In Section 3 the wireless sensor network layout problem is formulated. Section 4 presents and explains the algorithms employed. Then in Section 5 the experiments performed and the results obtained are shown. Finally the conclusions are drawn in Section 6.

2 Related Work

The positioning of nodes in a sensor network has received a notable attention in research. We present in this section a short review of the published research on this topic.

Zhang and Wicker [Zhang and Wicker 2005] study the positioning of sensors in a terrain from the point of view of data transmission. They divide the terrain into cells, then analyze how N sensors should be distributed among the cells, in a way that avoids network bottlenecks and data loss.

In their work [Biagioni and Sasaki 2003], Biagioni and Sasaki study different regular positioning methods for sensors: square, triangular and hexagonal grids. In each case they deduce the minimum number of sensors required to provide full coverage, and the resulting fault-tolerance, seen as the minimum number of nodes that have to be shut down in order to degrade the network coverage. They observe a tradeoff between node density and fault-tolerance, being the system with highest node density (thus highest number of nodes) the one with the highest fault-tolerance. In a similar approach, Kar and Banerjee [Kar and Banerjee 2003] propose systematic placing methods to ensure connected coverage to 2-dimensional regions and sets of points, that approach the minimum number of sensor nodes required and have polynomial execution times.

In [Dhillon and Chakrabarty 2003], Dhillon and Chakrabarty propose two greedy algorithms that select the locations for a sensor network with minimal number of nodes. They use a grid model for the terrain and consider a probabilistic coverage model for the sensors where the probability of coverage for any point by a given sensor decreases exponentially with its distance from the sensor. Their model allows them to include the effect of obstacles and terrain height as well as incorporate an *importance* factor that gives preference to the coverage of some part of the terrain. However, their model lacks an explicit method to handle network connectivity or energy optimization.

Other works study the performances of random node distributions in a terrain. Zhang and Hou [Zhang and Hou 2005] study the upper limit of the network lifetime with respect to a single node lifetime T ensuring α coverage (at least a part α of the terrain is covered by the sensor network), for a WSN with uniformly distributed nodes.

Network lifetime is regarded as the main objective in some works. Slijepcevic and Potkonjak [Slijepcevic and Potkonjak 2001] develop an algorithm that selects different sets of active nodes from a previously deployed WSN. Every set has to maintain full coverage and connectivity, and the number of sets has to be maximal in order to maximize the network lifetime. The proposed algorithm detects critical areas (minimum coverage zones) and forbids two nodes covering the same critical area from being in the same set. The technique is tested against simulated annealing, and obtains higher number of sets on average. With the same objective, Chen et al. [Chen et al. 2005] study the optimal number of sensor nodes and their positioning in a linear WSN using a greedy technique.

Heuristic methods have already been used to solve WSN problems involving network lifetime and coverage. Jourdan and de Weck solved an instance of WSN layout using a multi-objective genetic algorithm in [Jourdan and de Weck 2004]. In their formulation a fixed number of ten sensors has to be placed in order to maximize the coverage and the lifetime of the network. Djikstra's algorithm is repeatedly applied to the resulting topology to determine the number of rounds that can be performed provided each node has a predefined starting energy. Though the results obtained are encouraging, the small size of the network and the fact the the number of nodes is fixed instead of an optimizable value leave room for further research, as they state in their work.

We will contribute with this work to improve the state-of-the-art of the use of multi-objective metaheuristics for solving the WSN layout problem. Our aim is to provide an efficient solving method by comparing a set of state-of-the-art multi-objective techniques applied in the same scenario. We want to solve a new flexible instance in which, for the first time (to the best of our knowledge), both the number and positions of the sensors can be freely chosen, with full coverage of the sensor field guaranteed, and treating the energy efficiency and the overall cost of the network as independent optimization objectives. Besides this, our interest is to tackle complex instances in which the WSN size is in the same order of magnitude as real WSN, with several hundred nodes.

3 Wireless Sensor Network Layout Problem

A Wireless Sensor Network is a wireless network formed by sensor nodes. Each sensor node senses or monitorizes an area around itself called its sensing area.

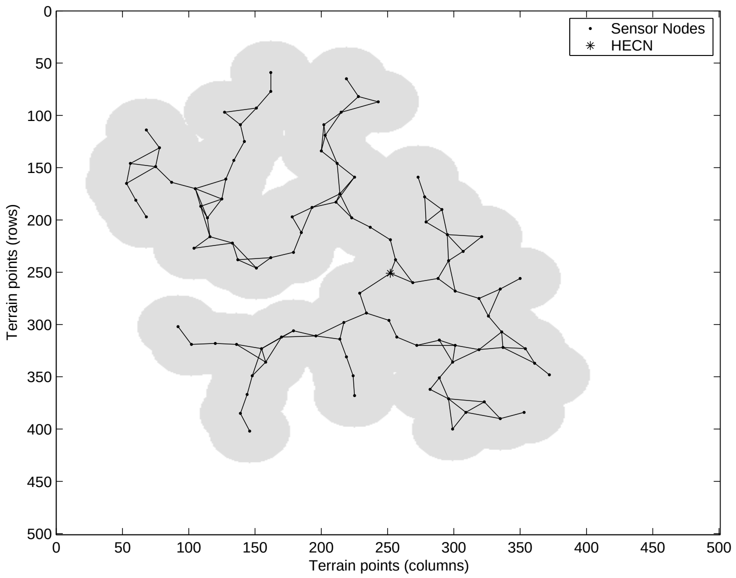

Figure 1: Wireless Sensor Network Example.

A parameter called *sensing radius* (R_{SENS}) determines the sensitivity range of the sensor node and thus the sensing area. The nodes communicate among themselves using wireless communication links. These links are determined by the parameter *communication radius* (R_{COMM}), which is the maximum distance at which two nodes can establish a link.

A special node in the WSN, the High Energy Communication Node (HECN), is the gateway for external access to the network. The administrator of the network gathers the measured data and sends commands through it. Therefore every sensor node in the network must have communication with the HECN. Since generally the communication radius is much smaller than the network size, direct links are not possible for peripheral nodes. A multihop communication path is established for those nodes that do not have the HECN within communication range using intermediate nodes as relays. An example WSN graphical representation is shown in Figure 3. In this illustration the HECN, located at the centre of the terrain, is represented using a star-shaped point and the standard nodes are represented using round dots. The network topology (communication links between nodes) are represented by lines and the covered terrain is shown in grey.

The Wireless Sensor Networks Layout problem amounts to providing the geographical position of the sensor nodes that form a WSN. A solution for this problem is a vector $\mathbf{x}$ of containing the 2D coordinates of the nodes. The coverage obtained by the network is a direct result of the number of nodes placed and the locations chosen for those nodes. In most –if not all– sensor network applications it is desireable to have as much coverage as possible.

Our aim is to obtain a full coverage network with minimum cost and maximum lifetime. The lifetime of a WSN is the period of time during which the network functions properly. As time passes, nodes will eventually run out of energy and stop operating, which results in a degradation of the network performance. The exact moment when the WSN stops functioning properly is subjective, but a broadly used measure for it is the time until the first node fails (*Time To First Failure*, TTFF) [Singh et al. 1998].

In our formulation, we use a discrete grid model for the terrain, where each point in the grid represents one square meter of the terrain. We assume a non-fixed amount of equal sensor nodes has to be placed in the terrain to provide it with full sensing coverage c (Eq. 1). The number of sensor nodes and their locations have to be chosen in a way that minimizes the energy spent in communications (Eq. 3) by the most loaded node in the network and the cost of the network which, in this case, is calculated as the number of deployed sensor nodes (Eq. 2). We minimize the load in the most loaded node of the network since this node constitutes the bottleneck of the network with respect to the network lifetime; the most loaded node will be the first node to run out of energy, hence determinating the network lifetime according to the TTFF criterion. The two objectives are opposed objectives, since the more nodes there are, the lower share of retransmissions each of them has.

$$c = 100 \cdot \left(\frac{Covered\ points(\mathbf{x})}{Total\ points} \right) = 100 \tag{1}$$

$$f_1(\mathbf{x}) = Length(\mathbf{x}) \tag{2}$$

$$f_2(\mathbf{x}) = Max \left(\{Energy\ consumed(x_i)\}_{i=1}^{f_1(\mathbf{x})} \right) \tag{3}$$

In order to determine the energy spent in communications by any node of the WSN, the number of transmissions performed is calculated. The WSN operates by rounds: in a round every node collects the data from its measurements and sends it to the HECN; between rounds the nodes are in a low-energy state. We consider a simple routing algorithm where every node transmits the information packets to its neighbors that is closest (in hop count) to the HECN, or the HECN itself whenever it is within communication range. When several neighbors are tied for the shortest distance from the HECN, the traffic is distributed evenly among them using random assignment. That is, if a node has n neighbors tied for shortest distance from HECN, each one receives $1/n$ of its traffic load. Therefore, every node has a traffic load equal to 1 (corresponding to its own sent data) plus the sum of all traffic loads received from neighbors that are farther from the HECN. Algorithm 1 shows the pseudocode for the calculation of the energy spent by every node.

Algorithm 1 Energy depletion

```
SetRoutingTree(WSN)
for node = from FirstNode(WSN) to LastNode(WSN) do
  Load(node) ← 0
end for
distance ← FurthestDistance(WSN)
while distance > 0 do
  for node ← from FirstNode(WSN) to LastNode(WSN) do
    if distance == Distance(node) then
      Load(node) ← Load(node) + 1
      relay neighbours ← 0
      for neighbour ← from FirstNeighbour(node) to LastNeighbour(node) do
        if Distance(neighbour) < Distance(node) then
          relay neighbours ← relay neighbours + 1
        end if
      end for
      for neighbour ← from FirstNeighbour(node) to LastNeighbour(node) do
        if Distance(neighbour) < Distance(node) then
          Load(neighbour) ← Load(neighbour) + Load(node)/relay neighbours
        end if
      end for
    end if
  end for
  distance ← distance - 1
end while
```

4 Optimization Techniques

In this section we present the optimization techniques employed to tackle the problem. These techniques are implemented using Paradiseo [Cahon et al. 2004], a white-box object-oriented framework dedicated to the flexible design of metaheuristics. In the last subsection, we will describe the operators we have specifically designed for this problem. These operators are common to all of the algorithms employed.

4.1 Nondominated Sorting Genetic Algorithm-II

Deb et al. proposed in [Deb et al. 2002] the second Nondominated Sorting Genetic Algorithm (NSGA-II) as a multi-objective technique that dealt with the main problems existing in the field: high computational complexity of nondominated sorting, lack of elitism and need of a sharing parameter specification. The authors fixed these problems by using a fast non-dominated sorting, an elitist Pareto dominance selection and a crowding distance method.

NSGA-II is based on a genetic algorithm. Its behavior can be seen in Algorithm 2. The differences between this algorithm and mono-objective GAs lie within the fitness assignment strategy.

In NSGA-II, the solutions are first sorted according to restriction fulfillment. Feasible solutions come first, then unfeasible solutions are sorted by increasing degree of constraint violation. Feasible solutions and every set of solutions with the same violation degree are the sorted according to Pareto dominance. This sorting is performed by successively extracting form the chosen subpopulation the set of non-dominated solutions (fronts). All the solutions in a front are given the same rank value, beginning at 0 for the first front extracted, 1 for the second and so on. This way, solutions can be sorted according to rank, starting at 0. Finally, within every group of solutions having the same rank, solutions are sorted according to the *crowding distance*. This criterion places first those solutions whose closest neighbors are farther, thus enhancing diversity.

4.2 Indicator-Based Evolutionary Algorithm

The IBEA was proposed by Zitzler and Künzli in [Zitzler and Künzli 2004]. This evolutionary algorithm does not employ the Pareto dominance concept –like NSGA-II does– but a performance indicator I instead. This indicator is a binary performance measure that can be used to compare two sets of non-dominated solutions. Using it, the goal of a multi-objective algorithm can be set to minimizing $I(A, S)$ where S is the Pareto set and A is the set obtained.

IBEA is an algorithm that employs an indicator as a sorting method for the solutions in the population. When the indicator is employed for this matter, it does not compare *sets* of non-dominated solutions, but single solutions instead. In this sense, every solution in the population is compared to every other solution and the results are added to get a single value. To enforce the dominance preference, the results of binary comparisons are not the indicators themselves, but instead the one shown in Eq. 4 (P is the population and k is a scaling factor that has to be greater than 0).

$$F(x^1) = \sum_{x^2 \in P \setminus \{x^1\}} -e^{-I(\{x^2\},\{x^1\})/k} \tag{4}$$

Paradiseo offers two indicators for the IBEA algorithm, both are defined in [Zitzler and Künzli 2004]: the *epsilon* indicator and the *hypervolume* indicator. The epsilon indicator $I_{\epsilon+}(A, B)$ gives the minimum distance by which a Pareto set approximation A needs to or can be translated in each dimension of the objective space such that it weakly dominates another approximation B. The

Algorithm 2 NSGA-II

```
t ← 0
Initialize(P_a)
while not EndingCondition(t,P_a) do
    Parents ← SelectionParents(P_a)
    Offspring ← Crossover(Parents)
    Mutate(Offspring)
    P_i ← Merge(P_a,Offspring)
    RankingCrowding(P_i)
    P_n ← ElitistSelection(P_i)
    t ← t + 1
    P_a ← P_n
end while
```

hypervolume indicator $I_{HD}(A, B)$ gives the volume of the space that is dominated by B but not by A.

4.3 Operators

This problem requires specific operators, mainly because of the special coding required for the candidate solutions: vectors of coordinates with varying lengths (the number of nodes, hence the number of coordinates in the vector, is an optimization objective). In the following subsections we will describe the initial solution generator, the mutation, and the crossover operators.

4.3.1 Initial Solution Generator

When a candidate solution is initialized, the number of nodes and their location have to be selected. In order to obtain good starting solutions for the algorithms, we decided to set the number of nodes in an initial solution using Eq. 5.

$$\#nodes = 4 \cdot \frac{Total\ Area}{\pi R_{SENS}^2} \tag{5}$$

This value equals four times the theoretical lower bound for the number of nodes necessary. This value was selected in such a way that initial solutions are likely to produce high coverage values, in order to help the optimization technique to quickly attain full coverage solutions.

Every node in the initial solution is assigned a random position selected as follows. A distance d is randomly selected from $[0, MinSize/2]$, where $MinSize$ is the minimum value between the x and the y sizes of the sensor field. An angle a is randomly selected from $[0, 360]$. The coordinates of the position of the node are then calculated with equations 6 and 7.

$$x = x_{HECN} + d \cdot \cos a \tag{6}$$

$$y = y_{HECN} + d \cdot \sin a \tag{7}$$

In these equations x_{HECN} and y_{HECN} are the coordinates of the HECN (which is located in the centre of the sensor field).

4.3.2 Mutation

We have designed two different kinds of mutation: the shift mutation, and the add-remove mutation. When mutation is applied to a candidate solution, either one or the other is used. The type of mutation used is selected at random, with relative probability of 60% for the shift mutation, and 40% for the add-remove mutation.

In the shift mutation, a randomly selected node of the WSN has its position modified by adding a random bounded translation movement (shifting) to it. To calculate the movement, a random distance d is selected between from $[0, R_{SENS}]$, and a random angle a is selected from $[0, 360]$. The shifting is then applied following equations 8 and 9.

$$x_{i+1} = x_i + d \cdot \cos a \tag{8}$$

$$y_{i+1} = y_i + d \cdot \sin a \tag{9}$$

If either x_{i+1} or y_{i+1} takes a value outside its permitted range (outside the sensor field), then the closest permitted value is assigned to the variable instead.

In the add-remove mutation, the size of the WSN is modified. This mutation adds a new sensor node with 50% probability, or removes a randomly selected node in the rest of the cases. When a new node is added, its location is selected using the same criterion as in the initial generator. The remove procedure is only performed if the WSN contains at least two nodes.

4.3.3 Crossover

We have adapted the two point crossover to the WSN case. This crossover can not be straightly used for this problem since the solution vectors will typically have different lengths. In our crossover, the indexes of the two points have to be selected from $[0, minsize-1]$, where $minsize$ is the length of the shortest vector selected for the crossover.

5 Experiments

In this work we solve an instance where a terrain of 500×500 meters has to be covered using nodes with coverage and communication radii equal to 30 meters. Since full coverage of the terrain is required, and no constraint handling mechanism is provided by Paradiseo, we have used a penalty method. In this method, a penalty function depending on the coverage is added to the two independent objectives, number of sensors and energetic load.

To define the penalty function we must first have a general idea of the upper bounds for the desired values. The terrain has an area of 250,000 square meters, and each sensor covers 2,827 square meters, meaning that in ideal conditions only 89 would be necessary. Now, these ideal conditions do not exist since they would imply that no overlap exists between any two nodes sensing areas, which is impossible due to their geometrical shape (circle). An example of solution that achieves full coverage of the region is a square grid formed by the sensors separated by 30 meters. Starting at the HECN, 250 meters have to be covered to each side of the terrain, requiring 8 sensors. Therefore the grid has 17 (8 + 8 + 1) rows and 17 columns, thus 289 sensors –including the HECN–. In this symmetrical configuration there are four nodes directly connected to the HECN, so the complete traffic of the network –288 messages per round– is evenly divided among them. This results in the most loaded nodes having a load of 72 messages. So this candidate solution obtains (288, 72).

All sensors need to be within communication range of at least another sensor, which results in an overlap between those two sensors's sensing areas (unless $R_{COMM} > 2R_{SENS}$). Therefore an upper bound for the possible coverage of a WSN using n nodes is shown in Equation 10.

$$Upper\ Limit = (n \cdot Coverage) - ((n-1) \cdot Overlap) \tag{10}$$

where we have:

$$Coverage = \pi \cdot R_{SENS}^2 \tag{11}$$

$$Overlap = R_{SENS}^2 \cdot (q - \sin(q)) \tag{12}$$

$$q = 2 \cdot \arccos\left(R_{COMM}/2R_{SENS}\right) \tag{13}$$

According to Eq. 10, the smallest amount of nodes required to produce full coverage (using $R_{COMM} = R_{SENS} = 30\ m$ and $Coverage = 250{,}000\ m^2$) is 94. However this upper bound for the coverage is not tight in a full-coverage scenario, since for each node in the WSN it takes into account the overlap with only one other node (its relay to the HECN). This implies the network has to be composed of a given number of non-overlaping spokes, which is not realistic since full coverage can not be achieved unless some overlap exists between the spokes. Therefore, the expected minimum number of nodes for full-coverage is higher than 94.

The solutions of the front will probably be in the order of magnitude of 10^2, so a suitable penalty function should compute in the same order of magnitude. Besides, we have identified three desirable features for the penalty function:

1. The function should be progressive: the more uncovered area, the larger the penalty function value. The function equals 0 when full coverage is attained.

2. A step at the origin is desireable, to stress the quality difference between *complete coverage* and *uncomplete coverage* –even for small values of uncoverage.

3. The penalty has to be large enough to counter the benefits (in the objectives) of violating the restriction.

The penalty function adds a value to the two objectives depending of the coverage achieved. We use Eq. 14 as the penalty function for this problem, where $u \in [0, 100]$ is the percentage of uncovered area ($u = 100 - c$).

$$Penalty(u) = \begin{cases} 0 & (u \quad = 0) \\ 100 & (0 \quad < u \leq 0.001) \\ 200 & (0.001 < u \leq 0.01) \\ 500 & (0.01 \quad < u \leq 0.1) \\ 1,000 & (0.1 \quad < u \leq 1) \\ 10,000 \cdot u & (1 \quad < u) \end{cases} \tag{14}$$

To solve the problem we apply a NSGA-II algorithm, two IBEA-like algorithms, and a MOEA. The MOEA algorithm is a generic genetic algorithm that employs the Pareto dominance concept by performing a NSGA-II-like nondominated sorting. The main difference between NSGA-II and MOEA is the selection method employed to select the next population: NSGA-II employs an elitist selection process (the best individuals are selected), while MOEA uses a roulette wheel, where each individual is assigned a selection probability according to its Pareto ranking (higher ranking values result in lower probabilities). We run 30 independent executions performing 100,000 solution evaluations for each of the algorithms. The non-dominated solutions obtained are returned at the end of each execution. We present in Table 1 the parametric configuration employed for the algorithms.

The two edges of the tradeoff between the number of nodes and the energy load in the most loaded node are plotted in Figure 2. In the left side we show the solution $Sol_{minnodes}$, which is the solution found with the minimum number of nodes (260 nodes, 123.5 load in the most loaded node). In the right side we

Table 1: Parametric configuration of the algorithms

MOEA		**NSGAII**		**IBEA**	
Population	100	*Population*	100	*Population*	100
Selection	*2-tourn.*	*Selection*	*2-tourn.*	*Selection*	*2-tourn.*
Offspring	100	*Offspring*	100	*Offspring*	100
Mutation	*shift addremove*	*Mutation*	*shift addremove*	*Mutation*	*shift addremove*
Mutation rate	0.8	*Mutation rate*	0.8	*Mutation rate*	0.8
Crossover	*dpx*	*Crossover*	*dpx*	*Crossover*	*dpx*
Crossover rate	0.8	*Crossover rate*	0.8	*Crossover rate*	0.8
Diversity	*crowding*	*Diversity*	*crowding*	*rho*	1.1
Replacement	*roulette*	*Replacement*	*elitist*	*kappa*	0.05

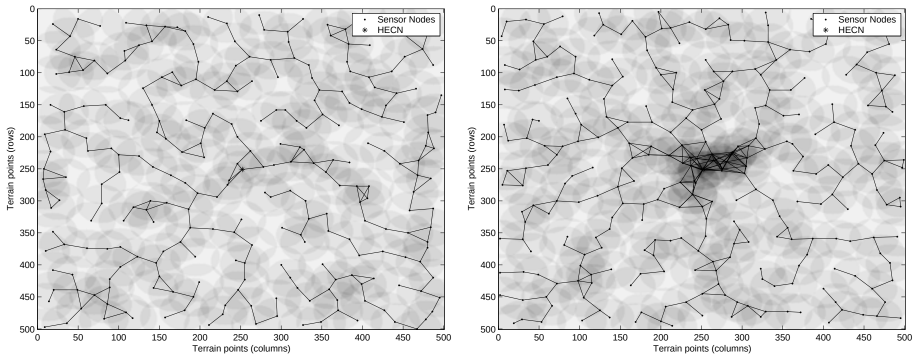

Figure 2: Example solutions obtained with MOEA: minimum number of nodes (left), minimum energy spent (right).

show the solution $Sol_{minload}$, which is the one with the smallest energy load in the network bottleneck (36 load in the most loaded node, 291 nodes).

Both solutions represented in Figure 2 produce full terrain coverage. The main difference between them can be found in the area close the HECN (in the center of the figure). The solution with the minimum number of nodes (left) has a low density of nodes, whereas the solution with minimum energy load (right) has a high density. All the information collected by the network has to be sent to the HECN, so the nodes that are close to the HECN have to retransmit all the information received from farther nodes. These nodes are thus likely to be the most loaded nodes of the network, since all the traffic load has to be supported by them. Assuming that the traffic load of the network is roughly similar in the two cases, the solution with higher node density around the HECN will have a larger lifetime, since the traffic load is shared among the nodes. Each of these nodes connects a spoke (or branch) to the HECN. If the network has N nodes and there are n such spokes, each spoke will approximately have N/n nodes;

the highest load in the network is then lower bounded by N/n. In the left side solution there are 260 nodes, 3 spokes, and the load is 123.5 (loosely bounded by $260/3 = 86.67$), in the right side solution there are 291, 8 spokes, and the load is 36 (very tightly bounded by $291/8 = 36.38$).

Table 2 shows the mean hypervolume of the fronts obtained by the different algorithms. Since in multi-objective optimization there is no unique manner to present the data obtained from the executions in a tabular form, we employ the hypervolume metric in order to measure the different algorithms obtained solutions. The hypervolume has been calculated as the union of the rectangular surfaces comprised between each point in the front and the top-right bound defined at (500, 250). The value is then normalized with respect to the rectangular area defined between to top-right bound, and the bottom-left bound at (250, 0); this value is selected in a way that the normalization value is the same for the two objectives (250), so that no objective is given preference. We also show in Figure 3 the best fronts obtained (those having the highest hypervolume value) for the different algorithms used.

Table 2: Experiment results for the four algorithms used

Algorithm	**MOEA**	**NSGA-II**	**IBEA$_\epsilon$**	**IBEA$_{HD}$**
Mean Hypervolume	**0.7388**	0.7306	0.7280	0.7189
Max. Hypervolume	0.7847	**0.7868**	0.7704	0.7654
Min. Sensors	**260**	262	262	265
Min. Load	**36**	41	41.5	41

Every algorithm tested was able to find a front of non-dominated feasible solutions (achieving full coverage of the sensor field) in every independent run. Our MOEA obtained slightly better results than the other three algorithms. The average hypervolume of the fronts obtained using the MOEA is 0.7388, while using NSGA-II it is 0.7306 and the using IBEA it is 0.7280 with the epsilon indicator, or 0.7189 with the hypervolume indicator.

We performed a statistical analysis to check the differences between the mean hypervolumes. This analysis is as follows. First a Kolmogorov-Smirnov test is performed in order to check whether the variables are normal or not. If they are, an ANOVA I test is performed, otherwise we perform a Kruskal-Wallis test. After that, we do a multiple comparison test. However, the differences are small and no statistical significance could be found.

In Fig. 3, we can notice that the number of nodes in the network (x axis) and the energy load in the most loaded node (y axis) are umbalanced in the fronts obtained. While the first has a range of approximately 20 units, the second has

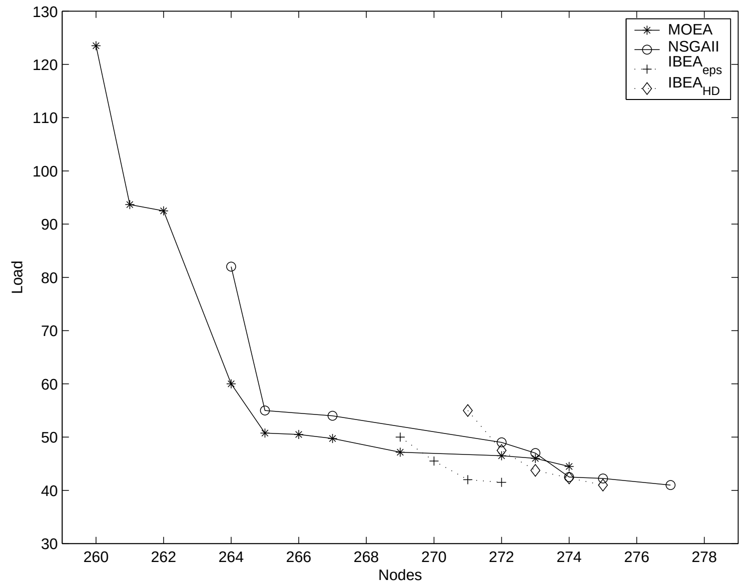

Figure 3: Best fronts obtained by the algorithms MOEA, NSGAII, IBEA$_\epsilon$ and IBEA$_{HD}$

around 80 units. A increase of 8% in the number of nodes leads to a decrease of 75% in the energy load due to transmissions. If we assume that this term dominates the overall energy consumption in the network (which is generally true), this would mean that a cost increase in the network of 8% would produce a network with four times as much lifetime. When the network designer desires to put emphasis on one of the objectives (cost or lifetime), additional constraints (e.g., maximum energy load) can be added as penalty functions to the objective functions in order to achieve better balanced fronts non-dominated solutions.

6 Conclusions

In this work several multi-objective metaheuristic algorithms have been used to solve the Wireless Sensor Network layout problem. This problem consists in providing the number and locations of the sensor nodes that form a WSN, so that full coverage of a given sensor field is achieved, and the lifetime and cost of the network are optimized. In our formulation of the problem, a 500×500 square meter area had to be fully covered using equal sensor nodes whose sensing and communication radii were set to 30 meters. The optimization objectives were to minimizes both the number of nodes deployed and the energy spent due to transmissions in the most loaded node. A standard Multi-Objective Evolutionary

Algorithm (MOEA), a Non-dominated Sorting Genetic Algorithm (NSGA-II) and two configurations of Indicator-Based Evolutionary Algorithms (IBEA$_\epsilon$ and IBEA$_{HD}$), implemented using the Paradiseo library, were applied to the problem instance. Every algorithm was able to find a set of non dominated feasible solutions in every independent execution, and the solutions obtained were found to be efficient by visualization analysis. The results obtained by MOEA were slightly better than those obtained by the other three techniques, although no statistical significance could be found in the difference between the algorithms. These results suggest that multi-objective metaheuristics are powerful and versatile methods to the solve the WSN layout problem. Additional constraints may be used to focus the search towards desired regions in terms of number of nodes and energy load. Further research should be driven in two major directions. The first line should lead to find algorithms that have staticaslly significant better performances than the rest (optimal techniques). The second line of research should lead to more realistic models for the problem, incorporating complete energy models for the nodes and terrain issues (height, obstacles, soil, etc.).

Acknowledgements

Authors are partially funded by the Spanish Ministry of Education and Science and by European FEDER under contract TIN2005-08818-C04-01 (The OPLINK project, http://oplink.lcc.uma.es), and by grant AP2005-0914 from the Spanish Government. Authors acknowledge founds from the INRIA-PERFOM 3+3 Mediterranean project.

References

[Akyildiz et al. 2002] Akyildiz, I.F., Su, W., Sankasubramaniam, Y., Cayirci, E.: "A survey on sensor networks". IEEE Communications Magazine (2002).

[Biagioni and Sasaki 2003] Biagioni, E., Sasaki, G.: "Wireless sensor placement for reliable and efficient data collection"; Proc. Hawaii Int. Conf. Sys. Sci. (Jan 2003).

[Cahon et al. 2004] Cahon, S., Melab, N., Talbi, E.: "Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics"; Journal of Heuristics, 10, 3 (2004), 357-380.

[Chen et al. 2005] Chen, Y., Chuah, C.N., Zhao, Q.: "Sensor placement for maximizing lifetime per unit cost in wireless sensor networks"; IEEE Military Comm. Conf., 2 (Oct 2005), 1097-1102.

[Culler et al. 2004] Culler, D., Estrin, D., Srivastava, M.: "Overview of sensor networks"; IEEE Computer, 37, 8 (2004), 41-49.

[Deb et al. 2002] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: "A fast and elitist multiobjective genetic algorithm: Nsga-ii"; IEEE Trans. Evo. Comp., 6, 2 (Apr 2002), 182-197.

[Dhillon and Chakrabarty 2003] Dhillon, S., Chakrabarty, K.: "Sensor placement for effective coverage and surveillance in distributed sensor networks"; Proc. IEEE Wirel. Comm. Netw. Conf. (2003), 1609-1614.

[Jourdan and de Weck 2004] Jourdan, D.B., de Weck, O.L.: "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm"; Proc. IEEE Semian. Vehic. Tech. Conf., 5 (2004), 2466-2470.

[Kar and Banerjee 2003] Kar, K., Banerjee, S.: "Node placement for connected coverage in sensor networks"; Proc. WiOpt 2003.

[Lédecze et al. 2005] Lédeczi, A., Nádas, A., Völgyesi, P., Balogh, G., Kusy, B., Sallai, J., Pap, G., Dóra, S., Molnár, K., Maróti, M., Simon, G.: "Countersniper system for urban warfare"; ACM Trans. Sen. Netw., 1, 2 (2005), 153-177.

[Liefooghe et al. 2007] Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.: "Paradiseo-moeo: A framework for evolutionary multi-objective optimization"; Lect. Notes in Comp. Sci. 4403, Springer, Berlin (2007), 386-400.

[Mladineo and Knezic 2000] Mladineo, N., Knezic, S.: "Optimisation of forest fire sensor network using gis technology"; Proc. 22^{nd} Int. Conf. Inf. Tech. Interf. (June 2000), 391-396.

[Nemeroff et al. 2001] Nemeroff J., Garcia L., Hampel D., DiPierro, S.: "Application of sensor network communications"; IEEE Military Comm. Conf., 1, 1 (2001), 336-341.

[Schwiebert et al. 2001] Schwiebert, L., Gupta, S.K.S., Weinmann, J.: "Research challenges in wireless networks of biomedical sensors"; Proc. 7^{th} Int. Conf. Mob. Comp. Netw., ACM, New York (2001), 151-165.

[Singh et al. 1998] Singh, S., Woo, M., Raghavendra, C.S.: "Power-aware routing in mobile ad hoc networks"; Proc. MOBICOM 1998, Dallas, Texas, 181-190.

[Slijepcevic and Potkonjak 2001] Slijepcevic, S., Potkonjak, M.: "Power efficient organization of wireless sensor networks"; IEEE Int. Conf. Comm. 2001, 2, 2, 472-476.

[Werner-Allen et al. 2006] Werner-Allen, G., Lorincz, K., Welsh, M., Marcillo, O., Johnson, L., Ruiz, M., Lees, J.: "Deploying a wireless sensor network on an active volcano"; IEEE Internet Computing, 10, 2 (2006), 18-25.

[Xu et al. 2004] Xu, N., Rangwala, S., Chintalapudi, K.K., Ganesan, D., Broad, A., Govindan, R., Estrin, D.: "A wireless sensor network for structural monitoring"; Proc. 2^{nd} Int. Conf. Embed. Netw. Sen. Sys., ACM, New York (2004), 13-24.

[Yuce et al. 2007] Yuce, M.R., Ng, S.W., Myo, Y.L., Khan, J.Y., Liu, W.: "Wireless body sensor network using medical implant band"; J. Med. Syst., 31, 6 (2007), 467-474.

[Zhang and Hou 2005] Zhang, H., Hou, J.C.: "On the upper bound of α-lifetime for large sensor networks"; ACM Trans. Sen. Netw., 1, 2 (2005), 272-300.

[Zhang and Wicker 2005] Zhang, X., Wicker, S.B.: "On the optimal distribution of sensors in a random field"; ACM Trans. Sen. Netw., 1, 2 (2005), 301-306.

[Zitzler and Künzli 2004] Zitzler, E., Künzli, S.: 'Indicator-based selection in multiobjective search"; PPSN 2004, 832-842.