Multi-objective Optimization Approach for Wireless Sensor Networks Deployment in Three Dimensional Environments

Abderrazak Daoudi[1], Youssef Kerfi[1], Imade Benelallam[2], El Houssine Bouyakhf [1]
[1]Mohammed V-Agdal University - Faculty of Science
Rabat, Morocco
[2]INSEA, Morocco
{daoudi, imade.benelallam}@ieee.org, y.kerfi@fsr.um5a.ma, bouyakhf@fsr.ac.ma

ABSTRACT: *With the emergence of wireless networking paradigm, several optimization problems are showing their usefulness to the efficient design of such networks. These problem are related, among others, to optimizing network connectivity, coverage and stability. The resolution of these problems turns out to be crucial for optimized network performance. In the case of Wireless Sensor Networks (WSN), such problems include computing placement of sensors, so that network performance is optimized. However, most optimization problems formulated for WSNs are variations of NP-hard optimization problems and are thus unlikely to be solvable in polynomial time. For instance, deployment of wireless sensor networks, to achieve a desired degree of coverage, are known for their hardness to solve to optimality. Therefore heuristics methods are used to near-optimally solve such problems. In this work we present the multiple coverage optimization problems in WSNs and a new algorithm methods based on the paradigm of constraint optimization problem (COP) for solving them optimally. We formulate the optimization problems using multi-objective optimization models. Thus, for the sensors placement, the multi-objective optimization problem is obtained consisting in the maximization of the number of sensors that monitor each point of the target area. We also present experimental results in order to evaluate the effectiveness of our approach for solving optimization problems in WSNs.*

Keywords: Constraint Optimization Problem (COP), Wireless Sensor Networks (WSN), Multi-objective coverage optimization

Received: 25 June 2012, Revised 8 August 2012, Accepted 19 August 2012

1. Introduction

Over the last few years, technological advances have led to the emergence of small devices that integrate sensors with highly efficient capabilities. Pervasive networks of such sensors open new perspectives for many potential applications, such as environment monitoring, military applications and biological detection [15]. Since,Wireless Sensor Networks are the key to gathering data from unattended and hostile environment. They have applications in a variety of fields such as environmental monitoring, military purposes and gathering sensing information in inhospitable locations [2].Coverage is one of the fundamental problems in sensor networks which sensing coverage characterizes the monitoring quality provided by a sensor network on a designated region and reflects how well a sensor network is monitored or tracked by sensors [20]. Consequently, coverage can be considered as the measure of quality of the service of a sensor network [22]. Coverage of target area can take many forms depending on the nature of applications. In less sensitive applications such as monitoring agricultural fields, we can design a coverage protocol such that each point in the target area is monitored by a single sensor. In this case, we speak of the 1-coverage or coverage simple. However, in critical applications such as military applications related to security, it is necessary to ensure coverage of each point in the target area by more than one sensor to involve tolerance fault and prevent the appearance of black points in these sensitive regions when a sensor stops working. These application are called k-coverage.

Another important point to consider in sensor networks is providing connectivity between sensor nodes. Without connectivity, nodes may not be able to coordinate effectively or transmit data back to base stations. Thus, combination of connectivity and coverage is an important concept in sensor networks [37].

In a large scale sensor network that the sensor nodes are random scattered, some redundant sensor nodes are needed to make sure a satisfying coverage on the sensing area. In sensor networks that sensor nodes can be set specially, we can reduce not only the redundancy nodes, routing request and maintenance overhead, power consuming but also expend the networks sensing range. So how to get the optimized node distribution is an important problem in the wireless sensor networks [43].

In this work, we introduce visibility analysis across space in addition to a new robust model for deployment optimization. Visibility analysis is used to generate visibility diagram of sensors in three dimensions environment. GIS-based approaches are used for optimizing the spatial coverage of WSN. New model is proposed for searching the most efficient sensors placement in order to maximize as much as possible the covered area. This approach can produce a robust and efficient solution that maximize cells covered by at least k sensors.

This paper is organized as follow. The next section reviews the associated literature on this topic. Section 2 reviews the background of the associated approach. Section 3 presents the coverage problem formulation. Thereafter, a sensor deployment models is suggested, followed by the presentation of the optimization algorithm. Finally, implementation details for this approach and application results are given.

2. Related works

This section surveys related works on Wireless Sensor Network coverage optimization using simplified assumptions. These works are different from our contribution in that it focuses on addressing the coverage problem partially in 2D environments. The model and solving techniques we develop are also quite different from previous approaches that deal with large cell reduced to one single point. Sensor deployment problems have been studied in a variety of contexts, and a number of techniques have been proposed for optimal deployment. Recently, many researchers have been investigating and developing techniques for achieving an optimal sensor deployment that maximizes the coverage of the target area and preserving sensor node connectivity. Huang et. al. [20] formulate the coverage problem as a decision problem, whose goal is to determine whether every point in the service area of the sensor network is covered by at least k sensors, where k is a predefined value. The sensing ranges of sensors can be unit disks or non-unit disks. They present polynomial-time algorithms, in terms of the number of sensors, that can be easily translated to distributed protocols. [34] presents an Integer Linear Programming formulation and evolutionary algorithm to find a configuration that maintains the coverage of the monitoring area, accomplishes the management of the network resources and minimizes energy consumption. [31] presents an optimal polynomial time worst and average case algorithm for coverage calculation by combining computational geometry and graph theoretic techniques (Voronoi diagram and graph search algorithms). Sensor coverage of the field are characterized by Maximal Breach Path and Maximal Support Path and these parameters can be used for future deployment or reconfiguration schemes for improving the overall quality of the service. [22] modeled the coverage problem as two sub-problems: floor plan and placement. Two sub-problems are combined into one optimization problem so that it can achieve the maximum possible coverage. [19] has formulated an optimization problem on sensor placement, where in a minimum number of sensors are deployed to provide sufficient coverage of the sensor field. Evolutionary computation and optimization techniques presented in these papers do not offer a solution to find an optimal coverage while preserving k-coverage and connectivity. Generally speaking, none of these approaches address the challenges of actual deployment.

3. Issues in Wireless Sensor Network Coverage

There are several factors that must be considered when developing a plan for coverage in a sensor networks. Many of these will be dependent upon the particular application that is being addressed. The capabilities of the sensor nodes that are being used must also be considered. Most researchers focus on a single deployment model but there are papers that attempt to develop a more general algorithm that can be used in many types of deployment.

3.1 Coverage Types

The first step in deploying a wireless sensor network is determining what it is exactly that you are attempting to monitor. Typically you would monitor an entire area, watch a set of targets, or look for a breach among a barrier. Coverage of an entire area otherwise known as full or blanket coverage means that every single point within the field of interest is within the sensing range of at least one sensor node. Ideally you would like to deploy the minimum number of sensor nodes within a field in order to achieve blanket coverage. This problem was addressed in [29] where theauthor proposes placing the nodes in a construct called an r-strip such that each sensor is located r distance away from the neighboring sensor where r is the radius of the sensing area. The strips can be then placed in an overlapping formation such that blanket coverage is achieved. The biggest problem with this solution is that it is impractical to try to deploy sensors in such a formation.

Target coverage refers to observing a fixed number of targets. This type of coverage has obvious military applications such as those covered in [4]. The authors in this paper did extensive tests to not only detect targets, but to classify and track them. The authors in [10], [13], [12], [42], and [41] attempt to maintain target coverage while conserving energy. The authors in discuss both blanket and target coverage in terms of energy efficiency.

Barrier coverage refers to the detection of movement across a barrier of sensors. This problem was defined as the maximal breach path in [32]. The authors in this study quantify the improvement in coverage when additional sensors are added to a network. Other papers such as [17] focus on algorithms in barrier coverage. A variation of barrier coverage known as sweep coverage is also discussed in [11] and [3]. Sweep coverage can be thought of as a moving barrier problem.

3.2 Deployment

A sensor network deployment can usually be categorized as either a dense deployment or a sparse deployment. A dense deployment has a relatively high number of sensor nodes in the given field of interest while a sparse deployment would have fewer nodes. The dense deployment model is used in situations where it is very important for every event to be detected or when it is important to have multiple sensors cover an area. Sparse deployments may be used when the cost of the sensors make a dense deployment prohibitive or when you want to achieve maximum coverage using the bare minimum number of sensors.

In most of the work studying coverage it is assumed that the sensor nodes are static, they stay in the same place once they are deployed. Newer sensor nodes have the ability to relocate after they are deployed, these are known as mobile nodes. The algorithm in [27] has each sensor node determining the location it needs to move to in order to provide maximum coverage. The authors perform several experiments to determine how well the network covers the area and the deployment time of the algorithm. The key weakness in this algorithm is that each node must be within the sensing range of another node in order to determine the optimal location it needs to move to, if a node is not seen by any other nodes then that node cannot determine its relative location. In the deployment algorithm of [33], each node will communicate with its neighbors and tell them to move away until they are at a distance which maximizes coverage while maintaining connectivity. The simulations run by the authors show a very high degree of coverage can be obtained within minutes of deployment. Actual sensors may not perform as well if they are not able to calculate the distance of their neighbors with the same precision as the nodes in the simulation. The method introduced in [3] aims to maximize coverage while minimizing sensor movement. The simulations run by the authors show the method does achieve excellent coverage with low amounts of movement but it does require a complex algorithm be run which may tax the sensor nodes. The authors in [38] design three separate deployment protocols that provide a high level of coverage with minimal movement in a short time. The simulations show that the protocols hold up with a limited amount of sensors but there are questions about how scalable the protocols are with larger numbers of sensors.

Sensor network nodes are deployed in an area by either placing them in predetermined locations or having the nodes randomly located. Dropping sensors from a plane would be an example of random placement. It is easier to develop a coverage scheme for deterministic placement of sensor nodes than for random placement. However in many deployments, it is either impractical or impossible to deploy sensor nodes in a deterministic way. Examples of deterministic and random placement is shown in figures 10 and 2. The simple construct in [27] is an example of a deterministic placement. A more sophisticated deterministic deployment method is given in [5]. The authors propose to arrange the sensors in a diamond pattern which would correspond with a Voronoi polygon. The pattern achieves four way connectivity from each of the nodes with full coverage when the communication range divided by the sensing range is greater than the square root of two. The authors are able to mathematically prove the validity of their pattern, however the pattern is not practical for actual deployment. It assumes that the sensing and communication ranges of every node are a perfect circle as well as the ability to place the sensors in exact locations. Random deployments of sensor nodes are usually dense deployments as well since it is necessary to deploy additional sensors in order to achieve coverage if the sensor nodes are stationary. Networks with mobile sensors usually start out with a random deployment and utilize

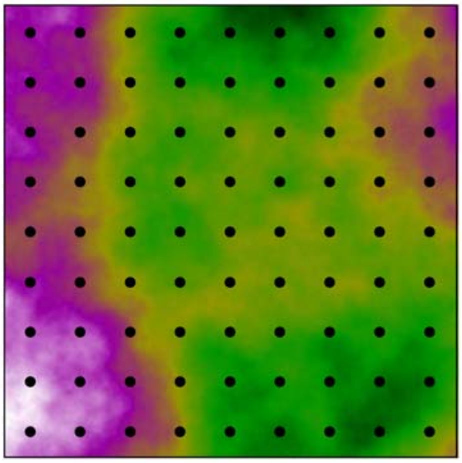

Figure 1. Deterministic Placement

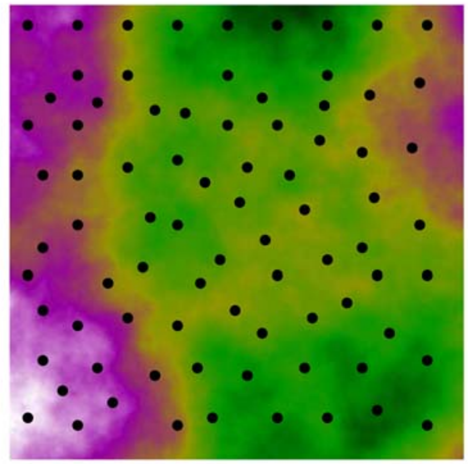

Figure 2. Random Placement

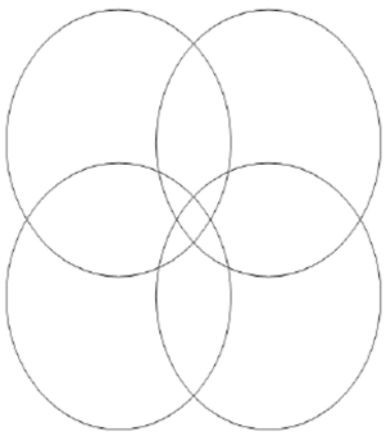

Figure 3. Homogeneous Sensors

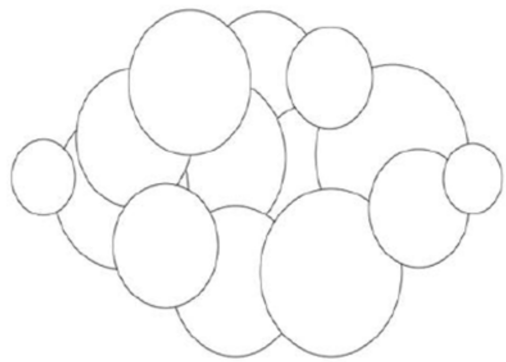

Figure 4. Heterogeneous Sensors

the mobility property in order to relocate to the optimal location. Most research with random deployments of sensor nodes regards the ability to maintain coverage while minimizing the amount of energy expended. This will be covered more closely in another part of the paper.

3.3 Node Types

The set of nodes that are selected for a sensor network can be either a homogeneous or heterogeneous group of nodes. A homogeneous group is a group in which all of the nodes have the same capabilities. A heterogeneous group is one in which some nodes are more powerful than other nodes. Usually you would have a smaller group of more powerful nodes known as cluster heads which would gather data from the less powerful nodes. Examples of homogeneous and heterogeneous nodes are given in figures 3 and 4.

A homogeneous set of nodes is required for the algorithms in [29] and [5]. Each of these solutions require the nodes to be placed at a precise distance in relation to each other that is dependent on the sensing ranges of every node being identical. The authors in [27] assume homogeneous sensors but repeat their experiments with different different uniform sensing ranges in order to prove the efficacy of their algorithm. Several algorithms for best coverage using homogeneous nodes are presented in [30].

Any algorithm that will work for a heterogeneous network will also work with a homogeneous network. Several papers attempt to prove their theories first with a homogeneous deployment then show that the findings will hold up for a heterogeneous deployment. In [32] the authors design a rectangular based coverage model using homogeneous sensors to monitor a barrier.

The authors do this by assuming a maximum and minimum sensing range and substituting these values into the theorem that was previously proven for homogeneous networks. The authors in [9] build an energy efficient network by using homogeneous sensors. This is then extended for heterogeneous networks. They do this by using a weighted Voronoi diagram.

3.4 Constraints

Perhaps the most important factor to consider in the development of a coverage scheme is that of energy constraints. Sensor nodes usually depend upon a battery for their energy source and in most deployments battery replacement is not feasible. It therefore becomes very important to conserve energy and prolong battery life. There are several methods available to do this. Placing unneeded sensors into a low energy sleep mode is a popular method to conserve energy. Another method is to adjust the transmission range so that the sensor nodes only use enough energy to transmit to a neighbor node. When sensors are arranged in a hierarchical network then cluster heads can be used to aggregate data and reduce the amount of information sent up to the sink. This will relieve some of the burden on the nodes that are along the transmission path and increase their lifetimes. Improving the efficiency of data gathering and routing is also used to conserve energy. If multiple sensor nodes are collecting the same information the network is expending energy unnecessarily. Eliminating the redundancy will allow the network to be more efficient. Optimizing the routing so that data is sent along the shortest path to the sink using the least number of nodes will conserve energy by lightening the routing burden on some nodes. By using less energy for routing data, coverage is helped by having the nodes' lifetimes extended. There is a great deal of research in the optimization of sensor routing but it is not directly related to the issue of coverage and will not be discussed further in this paper.

Cardei and Wu present a summary of different approaches to energy efficient coverage problems in [11]. The authors state that most work done in this field was in the theoretical realm at the time of the survey. Chen, Kumar, and Lai extend a barrier coverage protocol to improve energy efficiency. When a node detects adequate k-coverage in the area it will put itself into sleep mode. It will enter wakeup mode after a random period of time and perform another check. If the node is not needed then it will find out from the other nodes when and factor that into its calculation as to when it should wakeup again. The authors in and [13] conserve energy by turning off groups of nodes at a time. The authors in [40] introduce a new protocol in which the nodes can be in any of five different states. When a node wakes from the sleep state it will enter the listen state and wait for a beacon. After receiving the beacon the node determines if it should go back to sleep mode or go to the join state. From this state it will wait for its timer to expire and move to the active state unless it receives a message telling it to return to the sleep state. When the node is in the active state it is providing coverage to the area, it will remain in this state until it becomes ineligible at which point it moves to the withdraw state. Once in the withdraw state the node sets a timer and returns to the sleep state unless it receives a message telling it to return to the active state. The authors in [9] utilize a redundancy protocol in which the redundant sensors with the lowest energy levels are turned off. There may be several rounds of computation and sensors turned off until the optimal configuration is achieved.

A sensor node's coverage area is usually modeled as a disk in two dimensions or a sphere in three dimensions. Any point within

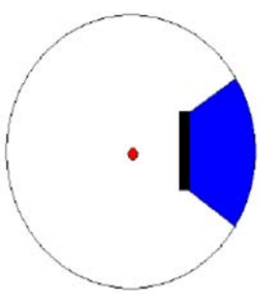

Figure 5. Effect of obstacle on one sensor

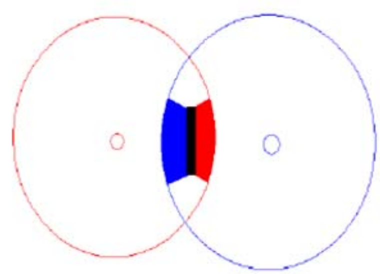

Figure 6. Effect of obstacle on two sensors

the area is assumed to be seen by the node. However in actual deployments the coverage area can be affected by obstacles. Examples of obstacles are walls and office equipment in indoor deployments, rocks and trees in outdoor deployments. Obstacles can absorb or reflect the RF signal put out by the node thereby rendering the area behind them invisible to the node. Examples of how an obstacle would affect one or two sensors are given in figures 5 and 6. The authors in [16] define different types of obstacles to be used in simulation environments. They specify several primitive shape such as circles, rectangles, and stripes

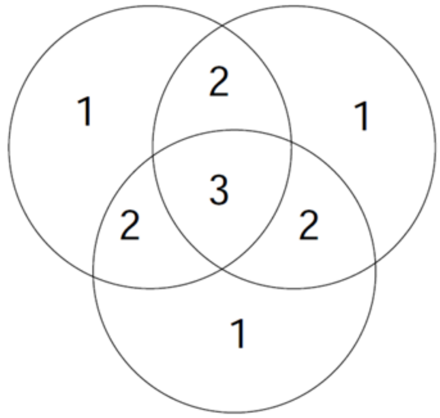

Figure 7. Heterogeneous Sensors

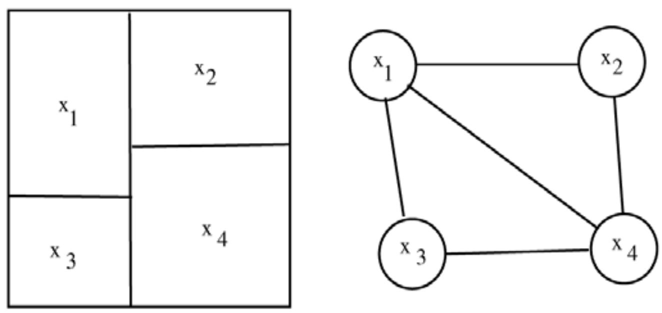

Figure 8. A simple map coloring problem and its representation as a constraint graph

that can be combined to simulate a real object in the sensing area. They run several experiments in a simulation environment to study the effects of obstacles on several routing protocols. They do not address the coverage problem in their work but their models or something similar could be used in coverage simulations. The effect of obstacles on coverage and connectivity is discussed by Wang, Hu, and Tseng in [39]. The authors assume a homogeneous set of nodes with a deterministic deployment and attempt to ensure coverage and connectivity with the minimum number of nodes. They divide the field of interest into smaller areas to determine where to deploy the sensors. The authors in [45] also consider forces exerted by obstacles. Knowledge of the terrain is needed to guide the initial random deployment in order to improve coverage.

Depending on the application an area may require that multiple sensors monitor each point in the field of interest. This constraint is known as k-coverage in which the k represents the number of nodes that watch each point. Requiring a k value of more than one will add complexity to the coverage algorithm. An example of k-coverage is illustrated in figure 7. The k-coverage constraint is closely related to the energy constraint in that most of the research that has been performed attempts to preserve k-coverage while minimizing the energy expended in the sensor nodes. This is the goal of the authors of [1], [44], and [24].

4. k-coverage

Sensor networks may consist of many different types of sensors such as seismic, low sampling rate magnetic, thermal, visual, infrared, acoustic and radar, which are able to monitor a wide variety of ambient conditions that include the following [23]

- temperature,
- humidity,
- vehicular movement,
- lightning condition,
- pressure,

- soil makeup,
- noise levels,
- the presence or absence of certain kinds of objects,
- mechanical stress levels on attached objects, and
- the current characteristics such as speed, direc- tion, and size of an object.

Sensor nodes can be used for continuous sensing, event detection, event ID, location sensing, and local control of actuators. The concept of micro-sensing and wireless connection of these nodes promise many new application areas. We categorize the applications into military, environment, health, home and other commercial areas. It is possible to expand this classification with more categories such as space exploration, chemical processing and disaster relief.

4.1 Military applications

Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT) systems. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military C4ISRT. Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile actions does not affect a military operation as much as the destruction of a traditional sensor, which makes sensor networks concept a better approach for battlefields. Some of the military applications of sensor networks are monitoring friendly forces, equipment and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage as- sessment; and nuclear, biological and chemical(NBC) attack detection and reconnaissance.

Monitoring friendly forces, equipment and ammunition: Leaders and commanders can constantly monitor the status of friendly troops, the condition and the availability of the equipment and the ammunition in a battlefield by the use of sensor networks. Every troop, vehicle, equipment and critical ammunition can be attached with small sensors that report the status. These reports are gathered in sink nodes and sent to the troop leaders. The data can also be forwarded to the upper levels of the command hierarchy while being aggregated with the data from other units at each level.

Battlefield surveillance: Critical terrains, approach routes, paths and straits can be rapidly covered with sensor networks and closely watched for the activities of the opposing forces. As the operations evolve and new operational plans are prepared, new sensor networks can be deployed anytime for battlefield surveillance.

*Reconnaissance of opposing forces and terrain:*Sensor networks can be deployed in critical terrains, and some valuable, detailed, and timely intelligence about the opposing forces and terrain can be gathered within minutes before the opposing forces can intercept them.Targeting: Sensor networks can be incorporated into guidance systems of the intelligent ammunition.

Battle damage assessment: Just before or after attacks, sensor networks can be deployed in the target area to gather the battle damage assessment data.

Nuclear, biological and chemical attack detection and reconnaissance: In chemical and biological warfare, being close to ground zero is important for timely and accurate detection of the agents. Sensor networks deployed in the friendly region and used as a chemical or biological warning system can provide the friendly forces with critical reaction time, which drops casualties drastically. We can also use sensor networks for detailed reconnaissance after an NBC attack is detected. For instance, we can make a nuclear reconnaissance without exposing a recce team to nuclear radiation.

4.2 Environmental applications

Some environmental applications of sensor networks include tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; macroinstruments for largescale Earth monitoring and planetary exploration; chemical/ biological detection; precision agriculture; biological, Earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire detection; meteorological or geophysical research; flood detection; biocomplexity mapping of the environment; and pollution study [6] [21] [8].

Forest fire detection: Since sensor nodes may be strategically, randomly, and densely deployed in a forest, sensor nodes can relay the exact origin of the fire to the end users before the fire is spread uncontrollable. Millions of sensor nodes can be deployed and integrated using radio frequencies/optical systems. Also, they may be equipped with effective power scavenging methods, such as solar cells, because the sensors may be left unattended for months and even years.

The sensor nodes will collaborate with each other to perform distributed sensing and overcome obstacles, such as trees and rocks, that block wired sensors' line of sight.

Biocomplexity mapping of the environment [14] A biocomplexity mapping of the environment requires sophisticated approaches to integrate in- formation across temporal and spatial scales. The advances of technology in the remote sensing and automated data collection have enabled higher spatial, spectral, and temporal resolution at a geometrically declining cost per unit area. Along with these advances, the sensor nodes also have the ability to connect with the Internet, which allows remote users to control, monitor and observe the biocomplexity of the environment.

Although satellite and airborne sensors are useful in observing large biodiversity, e.g., spatial complexity of dominant plant species, they are not fine grain enough to observe small size biodiversity, which makes up most of the biodiversity in an ecosystem. As a result, there is a need for ground level deployment of wireless sensor nodes to observe the biocomplexity. One example of biocomplexity mapping of the environment is done at the James Reserve in Southern California [14]. Three monitoring grids with each having 25-100 sensor nodes will be implemented for fixed view multimedia and environmental sensor data loggers.

Flood detection : An example of a flood detection is the ALERT system deployed in the US. Several types of sensors deployed in the ALERT system are rainfall, water level and weather sensors. These sensors supply information to the centralized database system in a pre-defined way. Research projects, such as the COUGAR Device Database Project at Cornell University and the DataSpace project at Rutgers, are investigating distributed approaches in interacting with sensor nodes in the sensor field to provide snapshot and long-running queries.

Precision Agriculture: Some of the benefits is the ability to monitor the pesticides level in the drinking water, the level of soil erosion, and the level of air pollution in realtime.

4.3 Health applications

Some of the health applications for sensor networks are providing interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital.

Telemonitoring of human physiological data: The physiological data collected by the sensor networks can be stored for a long period of time, and can be used for medical exploration. The installed sensor networks can also monitor and detect elderly people' s behavior. These small sensor nodes allow the subject a greater freedom of movement and allow doctors to identify pre-defined symptoms earlier. Also, they facilitate a higher quality of life for the subjects compared to the treatment centers. A "*Health Smart Home*" is designed in the Faculty of Medicine in Grenoble-France to validate the feasibility of such system.

Tracking and monitoring doctors and patients inside a hospital: Each patient has small and light weight sensor nodes attached to them. Each sensor node has its specific task. For example, one sensor node may be detecting the heart rate while another is detecting the blood pressure. Doctors may also carry a sensor node, which allows other doctors to locate them within the hospital.

Drug administration in hospitals: If sensor nodes can be attached to medications, the chance of getting and prescribing the wrong medication to patients can be minimized. Because, patients will have sensor nodes that identify their allergies and required medications. Computerized systems as described in [36] have shown that they can help minimize adverse drug events.

4.4 Home applications

Home automation: As technology advances, smart sensor nodes and actuators can be buried in appliances, such as vacuum cleaners, micro-wave ovens, refrigerators, and VCRs. These sensor nodes inside the domestic devices can interact with each other and with the external network via the Internet or Satellite. They allow end users to manage home devices locally and remotely more easily.

Smart environment: The design of smart environment can have two different perspectives, i.e.,human-centered and technology-centered. For human-centered, a smart environment has to adapt to the needs of the end users in terms of input/output capabilities. For technology-centered, new hardware technologies, networking solutions, and middleware services have to be developed. A scenario of how sensor nodes can be used to create a smart environment is described in [25]. The sensor nodes can be embedded into furniture and appliances, and they can communicate with each other and the room server. The room server can also communicate with other room servers to learn about the services theyoffered, e.g., printing, scanning, and faxing. These room servers and sensor nodes can be integrated with existing embedded devices to become self-organizing, selfregulated, and adaptive systems based on control theory models as described in [26]. Another example of smart environment is the "*Residential Laboratory*" at Georgia Institute of Technology. The computing and sensing in this environment has to be reliable, persistent, and transparent.

5. Constraint satisfaction Problem

A lot of problems in computer science, most notably in artificial intelligence, can be interpreted as special cases of the constraint satisfaction problem (CSP). Research work in the CSP area is focussed on the theoretical as well as the empirical investigation of methods to solve such problems [7]. While research activity is lively in the area, results are manifold already. Hence, representing and solving a given problem as a CSP pays off for complex problems where no efficient analytical solutions are known, as we will then be able to directly apply the latest findings in CSP technology.

Our definition of a CSP is based on [18]; it is called constraint network there, however.

Definition 5.1. (CSP) A constraint satisfaction problem (CSP) consists of :

- a finite set of variables $X=\{x_1,..., x_n\}$
- a finite set of discrete domains $D = \{D_1,...,D_n\}$, where Di represent the possible values for every $x_i \in X$,and
- a finite set of constraints $C = \{C_1,..., C_k\}$ on the variables of X.

We will still have to define the central notion of a constraint:

Definition 5.2. (Constraint) A constraint C_s on a tuple of variables $S = (x_1,..., x_r)$ is a relation on the product of these variables domains: $C_s \subseteq D_1\times ,..., \times D_r$.

The number r of variables a constraint is defined upon is called arity of the constraint.

A constraint comprises the values a variable is allowed to take with respect to other variables. Of special interest are the constraints with the arity two, the so-called *binary constraints*, as any arbitrary n-ary CSP can be transformed into a equivalent CSP with only unary and binary constraint [35]. Hence, most literature is only concerned with such *binary* CSPs, and we will consider the binary CSPs exclusively, too. A nice property of binary CSPs is that can easily be depicted as simple graphs, where the variables are denoted as nodes and the constraints as directed arcs between them; the direction of an arc gives the order of the variables in the corresponding constraint's tuple. The graph representation of a CSP is called *constraint graph*.

Now what is a solution of a CSP? To explain we need some more definitions:

Definition 5.3. (Instantiation of Variables) Let $X = \{x_1,..., x_n\}$ be a set of variables with their respective domains D_i, $i \in \{1,..., n\}$.Then any n-tuple $\Gamma = (a_1,...,a_n)$ where $a_i \in D_i$ denotes an instantiation of each variable x_i with the corresponding value a_i. We also write $\Gamma(x_i) = a_i$ for the value of x_i under an instantiation Γ.

Definition 5.4. (Satisfied Constraint) A constraint C_s on a tuple of variables $S = \{x_1,..., x_n\}$ is satisfied by an instantiation Γ if and only if $\Gamma(x_1),..., \Gamma(x_n) \in C_s$

Definition 5.5. (Solution of a CSP) An instantiation Γ is a solution of a CSP if it satisfies all the constraints of the problem.

The following examples illustrates the definitions.

Example 5.1. (*Map Coloring*) *In the map coloring problem, we want to color the regions of a map in a way that no two adjacent regions have the same color. The actual problem is that only certain limited number of colors is available. To cast a CSP out of this problem, we take the regions to be colored as the variables, and the set of colors as the variables domains. Let's say we have four regions as shown in figure* 8, *and only tree colors available. we get a set of variables* $X = \{x_1, x_2, x_3, x_4\}$ *with domains,* *e.g, $D_1 = D_2 = D_3 = D_4$ = fred, green, blueg. Now we just have to find a set of constraints to express our demand for adjacent regions having different colors. To ensure that $x1$ gets another color then x_2, we may define $C_{x1,x2} = \{(u, v) \in D_2 \times D_2 \mid u \neq v\}$, and just the like for $C_{x1,x3}$,$C_{x1,x4}$,$C_{x3,x4}$ and $C_{x2,x4}$. A solution of the CSP would than be Γ = (red, green, green, blue), for instance. Note that since our constraints are symmetric, the direction of the constraint arcs (i.e.,the order in a constraint's variable tuples) is insignificant in this example.*

In many other applications, decision combinatorial problems cannot be solved or we wish to find a solution to a CSP that is optimal with respect to certain criteria. To solve such problems, Constraints Optimization Problem (COP) approach is well indicated.

Definition 5.6. A constraint optimization problem (*COP*) is a tuple < X, D, C, R > where *X, D, C* are defined as in CSP, with a small difference:

• $C = \{C_1,...,C_p\}$: a finite set of hard constraints, and.

• $R = \{R_1,...,R_s\}$: a finite set of soft constraints, where Ri is function $D_{i1} \times ... \times D_{il} \rightarrow$ IR given the weight of each combination of values.

A solution of a COP is an assignment of values to all variables that satisfies all hard constraints and maximizes (or minimizes) the sum of all soft constraints. We call the sum of soft constraints the *objective function* of the COP.

Many optimized problems can be formulated as a COP. For example, soft graph coloring can be formulated as a COP where vertices are represented by variables, domains are sets of available colors, and soft constraints correspond to weighted edges of the graph.

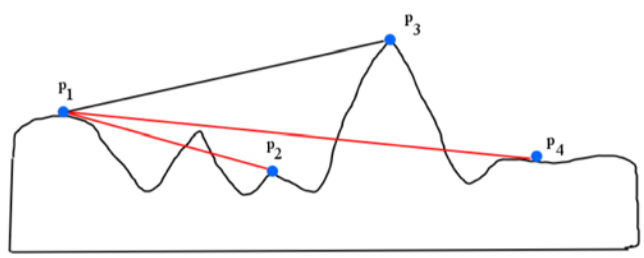

Figure 9. Line of sight and visibility

6. Visibility analysis in coverage problem

In this section definitions are given in order to introduce useful notations for coverage problem. The coverage problem is formulated with respect to visibility analysis.

The basic principle of visibility computation of digital elevation models (DEMs) is based on the line-of-sight (LOS) problem. If the visibility between two points is not blocked by the terrain or spatial objects, the points are visible mutually. Otherwise, they will be invisible. For example in Figure 9, p_1 is visible from p_3 but not visible from p_2. This is because line segment p_1p_3 (i.e.los ($\overline{p_1p_3}$) = true) is above the surface, but $\overline{p_1p_2}$ (i.e. $los(p_1, p_3) = false$) is blocked by the surface.

The area covered by a sensor relates to research known as visibility analysis. The observable area of a sensor is restricted by geographical features, like highlands buildings and trees, as well as the characteristics of a sensor. Such characteristics include relative elevation, vertical distance from a surface and sensor capabilities. Geographic Information Systems (GIS) are useful for delineating the visible area from a sensor given their computational geometry capabilities. We discretize the target area I into $m \times m$ points, as shown into Figure 10. A diagram of visibility V_i of a sensor placed on a point i can be defined as the set of points on the surface I that are visible from i extending out to some maximum distance R from the viewpoint. Formally:

$$\forall (i,j) \in I^2, v(i,j) = \begin{cases} 1 & \text{if } los(i,j) \text{ and } d(i,j) < R \\ 0 & \text{otherwise} \end{cases}$$

$$\forall j \in I \; V_i = \{ j \, / \, v(i,j) = 1 \}$$

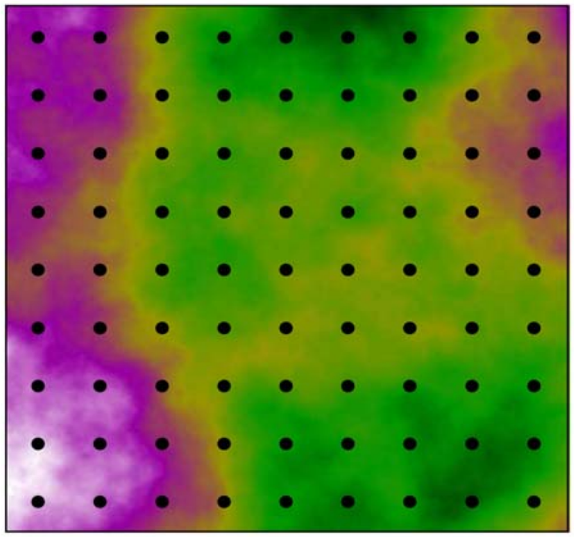

Figure 10. Surface I discretized into m × m points

7. Coverage model in Constraints Programming

In critical applications such as military applications related to security, it is necessary to ensure coverage of each point in the target area by more than one sensor, to involve tolerance fault and prevent the appearance of black points in these sensitive regions when a sensor stops working. The model presented in this section deal with this problem by introduced a good objective function to be optimized lexicographically. We consider n sensors to be deployed in a 3D environment. We associate with each sensor S_i a variable x_i which takes its values in $D(x_i) = \{1, 2, 3, ..., m \times m\}$, each value of this domain represent a allowed location that a sensor can take as shown in figure 10.

We have the following constraints:

• Hard constraints

- $d(x_i, x_j) >= d_{min}$: represents the minimum distance that should not be reached between a sensor placed in x_i and another placed in x_j, for failing to interference and overlap between the signals emitted and received by the sensors.

- $d(x_i, x_j) <= d_{max}$: represents the maximum distance that should not be exceeded between a sensor placed in x_i and another placed in x_j, for connectivity between the sensors.

- $los(x_i, x_j) = true$: represents visibility between a sensor placed in x_i and another placed in x_j, and ensures direct communication.

• Multi-objective function:

$$f(A) = (f_0(A), f_1(A), f_2(A), ..., f_n(A))$$

where A is a total assignment and fi is the number of cells covered by i sensors. The objective function mentioned above is represented by a mathematical function to optimize lexicographically. This structure allows each instantiation to calculate the number of points covered by 0, 1, 2,..., n sensors based on diagrams of visibility using the following soft constraint: V_{x_i} that represent the covered points when a sensor is deployed in a location from its domain. The lexicographically optimization allow us to minimize the number of points covered 0 time, otherwise, to minimize the number of non covered points, then it minimize the number of points covered one times, because, if we minimize the mono-objective $f_1(A)$, we increase automatically the other objective $f_2(A),...,f_n(A)$, as a result we maximize the number of points covered by more than one sensor. In general, if we minimize the mono-objective $f_1(A)$, we increase automatically the other mono-objectives $f_{i+1}(A),...,f_n(A)$.

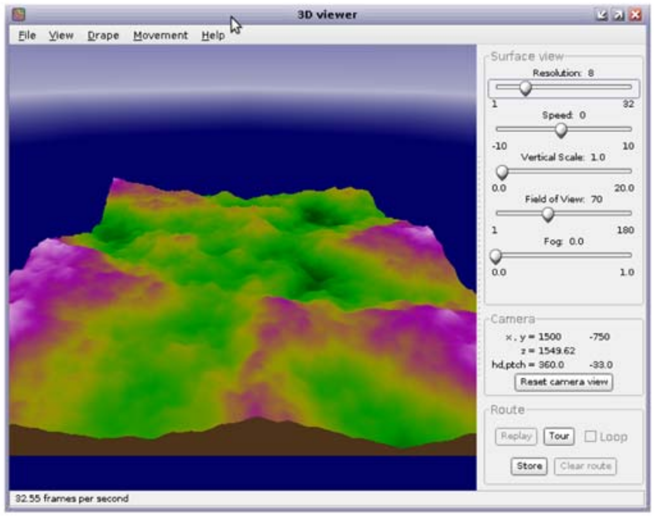

Figure 11. The geographical information system Landserf 3D viewer

---

Function BTWSN(*A, X, D, C, optimalCoverageVector*)
if *all variables are assigned in A* **then**
  CoverageVector = CalculateCoverageVector(A)
  if *Evaluate*(*CoverageVector, optimalCoverageVector*) **then**
    best solution $\leftarrow A$
    optimalCoverageVector $\leftarrow$ *CoverageVector*
else
  let x_i be an sensor not yet deployed
  sorting the domain D_{x_i} so that the first value selected has a maximum visibility
  foreach *possible location* $v \in D_{x_i}$ of x_i **do**
    $x_i := v$
    if $A \cup \{x_i := v\}$ *is consistent* **then**
      BTWSN($A \cup \{x := v\}$, *X, D, C, optimalCoverageVector*)
    else
      unassign x

Function CalculateCoverageVector(A):
return the civerage vector that content the number of
points covered by 0, 1, 2, 3...n sensors.
Function Evaluate(CoverageVector,optimalCoverageVector):
if *CoverageVector is lexicographically less than optimalCoverageVector* **then**
  return true
else
  return false

Algorithm 1. The BTWSN algorithm

8. Algorithm of resolution

For solving the model described above, we propose a Multi-objective optimization algorithm for WSN (BTWSN) presented in Algorithm 1. Where *X* is the set of all variables, *D* is the set of all domains, *C* is the set of all constraints, *A* is the current partial assignment, *optimalCoverageVector* is the optimal Decision Vector. In the first call to the function *BTWSN* the vector *optimalCoverageV ector* is set at infinity. BTWSN algorithm would perform a depth-first traversal through the search tree. The proposed algorithm is based on the backtracking search algorithm. Backtracking incrementally attempts to extend a partial solution that specifies consistent values for some of the variables, toward a complete solution, by repeatedly assigning a value for another variable consistent with the values in the current partial solution. A dead-end occurs when all the values of the current variable (i.e. the variable being instantiated) are rejected. In such a case, the variable that was instantiated before the current variable becomes uninstantiated. This process is called backtracking. For each total assignment found, a function objective is calculated(line 3). *BTWSN* algorithm would map every complete labeling of variables (solution) to a numerical vector. As the search proceeds, the *optimalCoverageVector* would be set to the value of the best solution found so far (line 5 and 6). The algorithm terminates when all possible assignments have been tested.

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
40	12	10	10	6	3	3	3	2	7	4

f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
38	14	10	10	6	3	3	3	2	7	4

Figure 12. Coverage vectors of the solution produced by our solver

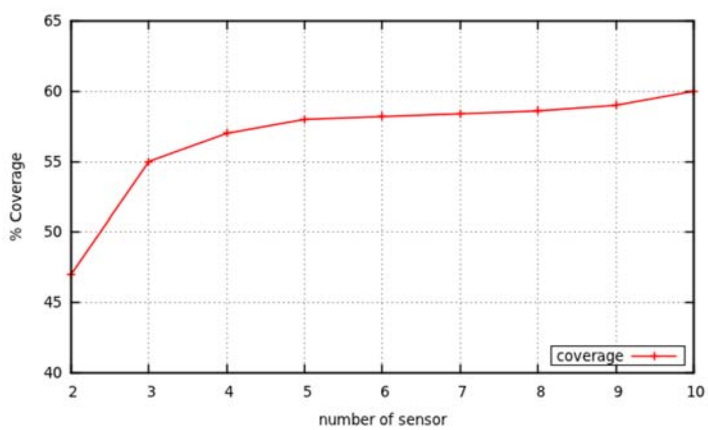

Figure 13. Area coverage with Number of Sensors varying

9. Experimental Results

Our approach described in the previous section is applied to deploy a set of sensors with different characteristics, such as the sensing radius r_s and minimum distance (d_{min}) and maximum distance (d_{max}) between two sensors. The sensors are deployed in 3D environments randomly generated with the geographical information system (GIS) landserf [28] shown in figure 11. The areas of deployment are discretized into 12×12 points.

The tables in the figure 2 represents the vectors of multi-objective function of our model obtained during the search of the optimal solution. The multi-objective function of the second solution (vector 2) is better than the multi-objective function of the first solution (vector 1), because it minimize f_0 (number of points not covered) and maximize f_1 (number of points covered by one sensor). The vector structure of the cost function allows the algorithm of resolution to maximize the area covered by more than one sensor.

In figure 3 we study the effects of coverage when the number of sensor varying between 2 and 10. As shown in this figure, when we reach a number of deployed sensors (4 sensors), the coverage percentage increases slowly, because the objective function described above in section 4, search first to monitor the target area by several sensors, so that we have this slow increase of the number of points covered by the deployed sensors.

10. Conclusion and perspectives

This paper has shown that sensor placement for supporting optimal coverage in 3D environments can be approached using constraints programming optimization combined with visibility analysis. Visibility analysis was used for calculating the coverage for each potential sensor location. We have obtained an optimal solution that maximize the area covered by a give number of sensors, and we have also proposed a solution to the problem of multiple coverage in WSN. It is also worth noticing that the BTWSN algorithm used in this paper does not perform any additional strategy, as it only computes the current cost of a assignment. A simple extension would be to propagate these particular soft constraints in the future variables in order to prune the search tree. We plane to include such as extension in the BTWSN algorithm and evaluate its impact. Further work is also needed to solve the problem of k-coverage in wireless sensor network, and we plan to develop a more complete performance evaluation.

References

[1] Zoë Abrams, Ashish Goel, Serge Plotkin. (2004). Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. *In*: *Proceedings of IPSN* 04, p. 424–432.

[2] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38, 393–422.

[3] Osmani, A., Dehghan, M., Pourakbar, H., Emdadi, P. (2009). Fuzzy-based movement-assisted sensor deployment method in wireless sensor networks. *In*: *First International Conference on Computational Intelligence*, *Communication Systems and Networks.*

[4] Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Gouda, M., Choi, Y., Herman, T., Kulkarni, S., Arumugam, U., Nesterenko, M., A., Vora, Miyashita, M. (2004). A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks* (Elsevier, 46, 605–634.

[5] Xiaole Bai, Ziqiu Yun, Dong Xuan, Ten H. Lai, Weijia Jia. Optimal patterns for four-connectivity and full coverage in wireless sensor networks.

[6] Ajay Bakre, Badrinath, B. R. (1995). I-tcp: Indirect tcp for mobile hosts. p. 136–143.

[7] Benelallam, I. (2010). *Approches complètes pour la résolution des problèmes DisCSPs et DCOPs*. PhD thesis, University Mohammed the fifth Agdal, Morocco.

[8] Nirupama Bulusu, Mani B. Srivastava, Stefano Soatto, John Heidemann, Mario Gerla, Deborah L. Estrin, Committee Chair. (2006). Self-configuring localization systems.

[9] Bogdan, C., Ananth Grama, Jan Vitek. Redundancy and coverage detection in sensor networks. *ACM Transactions on Sensor network* (ToSN, 2, 94–128.

[10] Mihaela Cardei, My T. Thai, Yingshu Li, Weili Wu. (2005). Energy-efficient target coverage in wireless sensor networks. *In*: *IEEE Infocom*, p. 1976–1984.

[11] Mihaela Cardei, Jie Wu. (2006). Energy-efficient coverage problems in wireless ad-hoc sensor networks.

[12] Mihaela Cardei, Jie Wu, Mingming Lu, Mohammad O. Pervaiz. (2005). Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. *In*: *Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications* (*WiMob*).

[13] Mihaela Cardei, Ding zhu Du. (2005). Improving wireless sensor network lifetime through power aware organization. A*CM Wireless Networks*, 11, 333–340.

[14] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod. (2001). Habitat monitoring: application driver for wireless communications technology. *In*: *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*.

[15] Krishnendu Chakrabarty, Senior Member, Sitharama Iyengar, S., Eungchun Cho, S. (2002). Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51,1448–1453.

[16] Mylonas Chatzigiannakis, G., Nikoletseas, S. (2006). Modeling and evaluation of the effect of obstacles on the performance of wireless sensor networks. *In*: *ACM/IEEE Simulation Symposium* (*ANSS* 06).

[17] Ai Chen. (2007). Designing localized algorithms for barrier coverage. *In*: *Proc. ACM MobiCom* ?07.

[18] Rina Dechter, Peter van Beek. (1997). Local and global relational consistency. *THE ORETICAL COMPUTER SCIENCE*, 173, 283–308.

[19] Dhillon, S.S., Chakrabarty, K. (2003). Sensor placement for effective coverage and surveillance in distributed sensor networks. *In*: *Wireless Communications and Networking*. WCNC. IEEE, 3 (3), p. 1609 –1614, march.

[20] Chi fu Huang, Yu-Chee Tseng. (2003). The coverage problem in a wireless sensor network. p. 115–121. ACM Press.

[21] Johannes Gehrke, Praveen Seshadri. (2000). Querying the physical world. *IEEE Personal Communications*, 7,10–15.

[22] Habib, S., Safar, M. (2007). Sensitivity study of sensors' coverage within wireless sensor networks. *In*: *Computer Communications and Networks*. ICCCN. *In*: *Proceedings of* 16th *International Conference on*, p. 876 –881, August.

[23] Sandra M. Hedetniemi, Stephen T. Hedetniemi, Arthur L. Liestman. (1988). A survey of gossiping and broadcasting in communication networks. *Networks*, 18 (4) 319–349.

[24] Mohamed Hefeeda. (2007). Randomized k-coverage algorithms for dense sensor networks. *In*: *Proceedings of INFOCOM* 2007 *Minisymposium,* p. 2376–2380.

[25] Herring, C., Kaplan, S. (2000). Component-based software systems for smart environments. *Personal Communications, IEEE*, 7 (5), 60 –61, oct.

[26] Herring, C., Kaplan, S. (2000). Component-based software systems for smart environments. *In*:*IEEE Personal Communications*, p. 60–61.

[27] Andrew Howard, Maja J Matari, Gaurav S Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *AUTONOMOUS ROBOTS, SPECIAL ISSUE ON INTELLIGENT EMBEDDED SYSTEMS*, 13:113–126, 2001.

[28] Landserf J. Wood. (2009). visualization and analysis of terrain models. http://www.landserf.org/.

[29] Koushik Kar, Banerjee, B. (2003). Node placement for connected coverage in sensor networks.

[30] Xiang-Yang Li, Peng-Jun Wan, Ophir Frieder. (2002). Coverage in wireless ad-hoc sensor networks, 2002.

[31] Megerian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. B. (2005). Worst and best-case coverage in sensor networks. *Mobile Computing, IEEE Transactions on*, 4 (1) 84 – 92, jan.-feb.

[32] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, Mani B. Srivastava. (2001). Coverage problems in wireless ad-hoc sensor networks. *In*: *IEEE INFOCOM*, p. 1380–1387.

[33] Sameera Poduri, Gaurav S. Sukhatme. (2004). Constrained coverage for mobile sensor networks. *In*: *IEEE International Conference on Robotics and Automation*, p. 165–172.

[34] Quintao, F. P., Nakamura, F. G., Mateus, G. R. (2005). Evolutionary algorithm for the dynamic coverage problem applied to wireless sensor networks design. *In*: *Evolutionary Computation*. The 2005 *IEEE Congress on*, 2, 1589 – 1596, sept.

[35] Francesca Rossi, Charles Petrie, Vasant Dhar. (1990). On the equivalence of constraint satisfaction problems. *In*: *Proceedings of the* 9th *European Conference on Artificial Intelligence*, p. 550–556.

[36] Barbara Sibbald. (2001). Use computerized systems to cut adverse drug events: report. *Canadian Medical Association Journal*, 164 (13) 1878–1878–a.

[37] Guiling Wang, Guohong Cao, Thomas F. La Porta. (2006). Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5, 640–652.

[38] GuilingWang, Guohong Cao, Thomas F. La Porta. (2006). Movement-assisted sensor deployment. p. 2469–2479.

[39] Wang, Y., Hu, C., Tseng, Y. (2005). Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. *In*: *WICON Proceedings*, p. 114 – 121.

[40] Guoliang Xing, Xiaorui Wang, Yuanfang Zhang, Chenyang Lu, Robert Pless, Christopher Gill. (2005). Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1, 36–72,

[41] Yang Yu, Viktor K. Prasanna. (2005). Energy-balanced task allocation for collaborative processing in wireless sensor networks. *In*: *Mobile Networks and Applications*, p. 1–2.

[42] Zhang, H., Wang, H., Feng, H. (2009). A distributed optimum algorithm for target coverage in wireless sensor networks. *In*: *Asia-Pacific Conference on Information Processing*.

[43] Ming-Chen Zhao, Jiayin Lei, Min-You Wu, Yunhuai Liu, Wei Shu. (2009). Surface coverage in wireless sensor networks. *In*: *INFOCOM*, IEEE, p. 109 –117, April.

[44] Zongheng Zhou, Samir Das, Himanshu Gupta. (2004). Connected k-coverage problem in sensor networks. *In*: *Proceedings of the International Conference on Computer Communications and Networks* (*IC*3*N*), p. 373– 378.

[45] Yi Zou,Krishnendu Chakrabarty. (2004). Sensor deployment and target localization in distributed sensor networks. *ACM Transactions* on Embedded Computing Systems, 3 (1) 61–91.