

UNIVERSITÀ DELLA CALABRIA



Dipartimento di ELETTRONICA,
INFORMATICA E SISTEMISTICA

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
XXIV ciclo

Tesi di Dottorato

Algorithms and techniques towards the
Self-Organization of Mobile Wireless Sensor,
Robot and UAV Networks

Carmelo Costanzo



UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
Ciclo XXIV

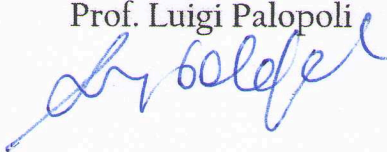
Tesi di Dottorato

Algorithms and techniques towards the
Self-Organization of Mobile Wireless Sensor,
Robot and UAV Networks

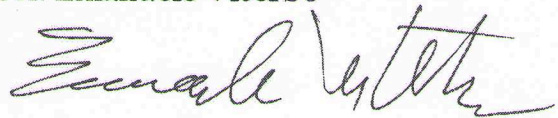
Carmelo Costanzo

Carmelo Costanzo

Coordinatore
Prof. Luigi Palopoli



Supervisore
Prof. Emanuele Viterbo



DEIS - DIPARTIMENTO DI ELETTRONICA, INFORMATICA E SISTEMISTICA
Novembre 2011

Settore Scientifico Disciplinare: ING-INF/03

Carmelo Costanzo

Algorithms and techniques
towards the Self-Organization of
Mobile Wireless Sensor, Robot
and UAV Networks

November 28, 2011

*to my family
and to my fiancée, Francesca*

God does not throw dice. (Albert Einstein)

God plays dice with the universe, but they're loaded dice. (Joseph Ford)

*Not only does God definitely play dice, but He sometimes confuses us by throwing
them where they can't be seen. (Stephen Williams Hawking)*

Stop telling God what to do with his dice. (Niels Bohr)

Contents

Introduction	1
1 Wireless Sensor, Robot and UAV Networks as a Self-Organized Networked System	3
1.1 Introduction	3
1.2 Wireless Sensor, Robot and UAV Networks	3
1.2.1 Robot networks	5
1.2.2 UAV networks	6
1.3 Performance parameters	8
1.3.1 Energy consumption and lifetime	8
1.3.2 Coverage and connectivity	9
1.3.3 Throughput and delay	11
1.4 Mobility	11
1.4.1 Advantages and Limitations of Controlled Mobility	12
1.4.2 Challenges layer by layer	17
1.5 Self-organizing Systems (SOS)	20
1.6 Conclusion	23
2 Modelling and Solving Optimal Placement problems	25
2.1 Introduction	25
2.2 State of art	27
2.3 Problem Statement and Mathematical Formulation	28
2.3.1 Model 1: Maximization of the total residual energy	30
2.3.2 Model 2: Maximization of the minimum residual energy	30
2.3.3 Model 3: Maximization of the number of nodes with residual energy above a threshold	31
2.3.4 Model 4: Minimization of the total travelled distance ..	31
2.3.5 Model 5: Minimization of the maximum travelled distance	32
2.3.6 Centralized Placement Algorithm	32
2.3.7 A routing technique	34

2.4	Distributed Heuristic Strategies	35
2.5	Computational experience	38
2.6	Conclusions	43
3	Controlled Mobility: algorithms and protocols	45
3.1	Introduction	45
3.2	Optimal placements and Mobility Schemes for Improving Energy Efficiency	46
3.2.1	State of art	46
3.2.2	Optimal placements	48
3.2.3	Real movement	53
3.2.4	Virtual movement	63
3.2.5	Results and Discussion	74
3.3	Controlled mobility assisted Routing protocol	83
3.3.1	State of art	84
3.3.2	Practical Applications of Controlled Mobility	86
3.3.3	A new Routing Protocol based on Controlled Mobility (RPCM)	87
3.3.4	Simulation and Results	91
3.3.5	Conclusion	97
3.4	A Discrete Stochastic Process for Coverage Analysis	98
3.4.1	State of art	99
3.4.2	Markov Chain and Coverage Metrics	100
3.4.3	Transition Probabilities	103
3.4.4	Coverage Performance and its Discussion	106
3.4.5	Conclusions	110
3.5	Conclusions	110
4	Bio-Inspired mechanism for Self-Organization	111
4.1	Introduction	111
4.2	Distributed Algorithm to Improve Coverage for Mobile Swarms of Sensors	112
4.2.1	Introduction	112
4.2.2	Related Work	113
4.2.3	Proposed Algorithms	114
4.2.4	The Simulation Environment	117
4.2.5	Conclusion	123
4.3	Nodes self-deployment for coverage maximization using an evolving neural network	123
4.3.1	State of art	125
4.3.2	A Genetic Algorithm for Nodes Self-Deployment	127
4.3.3	Performance evaluation	132
4.3.4	Conclusion and future direction	141
4.4	Evolving neural networks for self-control mobility to address coverage problem	141

4.4.1	Proposed Model	142
4.4.2	Performance evaluation	145
4.4.3	Conclusion	150
4.5	Conclusion	151
5	A case of study: Efficient Coverage for Grid-Based Wireless Sensor Networks	153
5.1	Introduction	153
5.2	State of art	154
5.3	Square Grid Networks.....	155
5.4	The optimization model	156
5.5	Coverage Techniques	159
5.5.1	Genetic Algorithm	159
5.5.2	Virtual Forces Algorithm (VFA).....	160
5.5.3	A Map-assisted coverage heuristic	160
5.6	Performance Evaluations	162
5.6.1	Zone of Interest	162
5.7	Conclusion	170
	Conclusion	171
	References	175

Introduction

This thesis involves the study of mobile wireless sensors, robots and UAV networks, in particular it deals with techniques and algorithms that allow the network to exploit nodes mobility through the optimization of some network's parameters, such as coverage and energy consumption.

Wireless sensor networks (WSN) are multi-hop high density networks without any specified topology, that sense the environment, collect data and communicate within the neighborhood. When a node is equipped with mechanic devices that allow it to move and specific mechanisms that permit to control and coordinate individual's and group's behavior, it belongs to a Robot Network or to a UAV network, in case of aerial vehicles. More general terms exist in literature that try to define such networks. An interesting definition that could embrace these new kinds of networks is the concept of Wireless Sensor and Actor Network (WSAN) in [40].

The network issues have been addressed using different approaches: from the theoretical studies aimed at finding the maximum achievable performance benchmarks through the introduction of appropriate optimization models, to the proposal of distributed heuristics and more realistic communication protocols, and to the use of biology-inspired mechanisms, such as genetic algorithms (GA), particle swarm optimization (PSO) and neural networks (NN). The purpose of this type of approach is to move in the direction of networks that are able to self-organize by adapting to different environmental conditions and dynamic as well as complicated scenarios.

The remainder of this thesis is organized as follows. In Chapter 1, the background on Wireless Mobile Networks is given. Optimization models for the minimization of the energy consumption and of the distances travelled by the nodes are presented in Chapter 2, distributed heuristics and communication protocols in Chapter 3 and finally several bio-inspired approaches in Chapter 4. Chapter 5 concludes this work with a case of study where all previous mentioned approaches are used and compared.

Wireless Sensor, Robot and UAV Networks as a Self-Organized Networked System

1.1 Introduction

The aim of this chapter is to introduce the concept of Wireless Mobile Networks (Wireless Sensor, Robot and UAV Networks) by emphasizing the features of this kind of networks, the main parameters used to measure the Quality of Service (QoS) and by proposing the mobility as a primitive of the network. Using coordination and cooperation mechanisms in an appropriate way, such kind of network could be considered as a Self-Organizing System (SOS) where mobility is exploited in a way to achieve self-deployment and optimize some network's performance parameters such as coverage and energy consumption.

The remainder of the chapter is organized as follows. In section 1.2 background on Wireless Sensor Networks is given with focus on different features of sensor, robot and UAV networks. Main performance parameters useful to give a measure of effectiveness of the networks are investigated in section 1.3. The analysis of mobility as control primitive of such communication devices composing the network is described in section 1.4. Finally in section 1.5 will be introduced the innovative concept of Self-Organizing System with particular focus on his potentiality on network issues.

1.2 Wireless Sensor, Robot and UAV Networks

A wireless sensor network (WSN) consists of low-cost, low-power, multifunctional, autonomous sensor nodes deployed either randomly or according to some predefined statistical distribution, over a geographic region of interest to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location usually a sink node with more energy and processing and communication capabilities. The development of wireless sensor networks was motivated by military applications such

as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, transportation, entertainment, crisis management, homeland defense, home automations and smart spaces.

The WSN is built of nodes, from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node consist of communication capabilities through a radio transceiver with an internal antenna or connection to an external antenna, processing capabilities through a microcontroller, possibilities of storing data using different type of memories and finally an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from some centimeters down to the size of a grain of dust. Also the cost of sensor nodes is variable in the ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, mobility features, computational speed and communications bandwidth.

A sensor node by itself has severe resource constraints, such as low battery power, limited signal processing, limited computation and communication capabilities, and a small amount of memory; hence it can sense only a limited portion of the environment. However, when a group of sensor nodes collaborate with each other, they can accomplish a much bigger task efficiently. One of the primary advantages of deploying a wireless sensor network is its low deployment cost and freedom from requiring a messy wired communication backbone, which is often infeasible or economically inconvenient.

Due to these constraints, resource management is of critical importance to these networks. Sensor nodes are scattered in a sensing field with varying node densities. Typical node densities might vary from nodes $3m$ apart to as high as $20nodes/m^3$. Each node has a sensing radius within which it can sense data, and a communication radius within which it can communicate with another node. Each of these nodes will collect raw data from the environment, do local processing, possibly communicate with each other in an optimal fashion to perform neighborhood data or decision fusion (aggregation), and then route back those aggregated data in a multi-hop fashion to data sinks, usually called the base-stations, which link to the outside world via the Internet or satellites. Since an individual node measurement is often erroneous because of several factors, the need for collaborative signal and information processing is critical.

One important criterion for being able to deploy an efficient sensor network is to find optimal node placement strategies. Deploying nodes in large sensing fields requires efficient topology control. Nodes can either be placed manually at predetermined locations or be dropped from an aircraft. However, since the sensors are randomly scattered in most practical situations, it is difficult to find a random deployment strategy that minimizes cost, reduces computation and communication, is resilient to node failures, and provides a high degree

of area coverage. The notion of area coverage can be considered as a measure of the quality of service (QoS) in a sensor network, for it means how well each point in the sensing field is covered by the sensing ranges. Once the nodes are deployed in the sensing field, they form a communication network, which can dynamically change over time, depending on the topology of the geographic region, inter-node separations, residual battery power, static and moving obstacles, presence of noise, and other factors.

Routing protocols and node scheduling are two other important aspects of wireless sensor networks because they significantly impact the overall energy dissipation. Routing protocols involve primarily discovery of the best routing paths from source to destination, considering latency, energy consumption, robustness, and cost of communication. Conventional approaches such as flooding and gossiping waste valuable communication and energy resources, sending redundant information throughout the network. In addition, these protocols are neither resource-aware nor resource-adaptive. Challenges lie in designing cost-efficient routing protocols, which can efficiently disseminate information in a wireless sensor network using resource-adaptive algorithms. On the other hand, node scheduling for optimal power consumption requires identification of redundant nodes in the network, which can be switched off at times of inactivity.

Synthesizing the main characteristics of a WSN include:

- power consumption and processing capabilities constrains;
- dynamic network topology;
- heterogeneity of nodes;
- mobility of nodes;
- scalability to large scale of deployment;
- ability to cope with node failures;
- ability to withstand harsh environmental conditions.

More recently the communication devices composing the WSN are not only equipped with sensors but also with mechanical devices that allows the movement on the ground or the fly of such nodes. In this contest we will refer to these kinds of networks respectively as Robot networks and UAV networks. In literature such networks with more intelligence and abilities are reported as Wireless Sensor and Actor Networks (WSANs) [40]. More specific details about these networks will be given in the followings subsections.

1.2.1 Robot networks

The technological development of the last decade in robots, computing and communications has led to envisage the design of robotic and automation systems consisting of networked vehicles, sensors, actuators and communication devices. These developments enable researchers and engineers to design new

robotic systems that can interact with human beings and other robots in a cooperative way. This new technology has being denominated “Network Robot Systems” (NRS) and includes the following elements [39]:

- Physical embodiment: any NRS has to have at least a physical robot which incorporates hardware and software capabilities;
- Autonomous capabilities: a physical robot must have autonomous capabilities to be considered as a basic element of a NRS;
- Network-based cooperation: the robots, environment sensors and humans must communicate and cooperate through a network;
- Environment sensors and actuators: besides the sensors of the robots, the environment must include other sensors, such as vision cameras and laser range finders, and other actuators, such as speakers and flickers;
- Human-robot interaction: in order to consider a system as NRS, the system must have a human-robot related activity.

The European study group Research Atelier on Network Robot Systems inside of EURON II has given the following interesting definition of NRS: “A Network Robot System is a group of artificial autonomous systems that are mobile and that make important use of wireless communications among them or with the environment and living systems in order to fulfill their tasks”. Network Robot Systems (NRS) call for the integration of several fields: robotics, perception (sensor systems), ubiquitous computing, artificial intelligence, and network communications. Some of the key issues that must be addressed in the design of Network Robot Systems are cooperative localization and navigation, cooperative environment perception, cooperative map building, task allocation, cooperative task execution, human-robot interaction, network tele-operation, and communications. The topic Network Robot Systems transcends conventional robotics, in the sense that there exists, for these type of distributed heterogeneous systems, an interrelation among a community of robots, environment sensors and humans. Applications include network robot teams (for example to play soccer), human-robot networked teams (for example a community of robots that assist people), robots networked with the environment (for example for tasks on urban settings or in space applications) or geminoid robots (a replication of a human with own autonomy and being partially tele-operated through the network).

1.2.2 UAV networks

Advances in control engineering and material science and low cost and high performance of commercial wireless equipment made it possible to develop small-scale unmanned aerial vehicles (UAVs) equipped with cameras, sensors and communication devices. The technology originates from military applications, recently, have also been offered this kind of products also for the commercial market and have gained much attention. In civil applications UAVs

can act for example as relays between ground stations that could not otherwise communicate due to distance or obstructed line of sight. Multiple UAVs could simultaneously detect, record and track wildfires. Last but not least, UAV networks can be deployed on demand to create an instant communication infrastructure for example to facilitate temporary hot spots and compensate network outages in case of public events and emergencies. Today such UAVs are used specially for aerial imaging, police and fire rescue operations, and military missions.

An UAV network can be regarded as an autonomous system that flies in the air, senses the environment, and communicates with the ground station. Typically it is controlled by a human operator by remote control. Despite these advances, the use of a single UAV has severe drawbacks, demanding for a system in which several UAVs fly in a formation and cooperate in order to achieve a certain mission. Potential opportunities and benefits of multiple cooperating UAVs include the following [38]:

- a single UAV cannot provide an overall picture of a large area due to its limited sensing range, limited speed, and limited flight time. Furthermore, it has only a limited view onto the ground due to buildings, trees, and other obstacles. A formation of UAVs can cover a much larger area. In addition, multiple views on a given scene, taken by different UAVs at the same time instant, can help to overcome the problem of occlusion;
- by intelligently analyzing different views, the image quality can be improved and even depth information can be computed, leading to a three-dimensional model of the environment;
- using GPS system based navigation and sophisticated on-board electronics that lead to high stability in the air, by communicating each other the direction of provenience, when a meeting occur, is possible to achieve full coverage in a minimum time [151];
- last but not least, an aerial imaging system working with a multitude of UAVs can be made more robust against failures and allows a certain level of task sharing among the UAVs.

A vision for the future is to have an aerial imaging system in which UAVs will build a flight formation, fly over an area of interest, and deliver high-quality sensor information such as images or videos. These images and videos are communicated to the ground, fused, analyzed in real-time, and finally presented to the user. The main tasks for collaborative UAVs are as follows:

- Flying in a structured and controlled manner over a predefined area;
- Sensing the environment, i.e., taking pictures, recording video data, and possibly fuse it with the data from other sensors, e.g., infrared sensors and audio sensors;
- Analyzing sensor data, either off-line at the ground station or on-line, during flight, and in a collaborative manner and presenting the results to the user;

- Processing the sensor data on-board during flight, performing object detection, classification and tracking.

Adding properly specific mechanisms for autonomous control flying, and cooperation with other UAVs, such a UAV formations behaves as a Self-organizing system such as swarm of birds.

1.3 Performance parameters

In order to evaluate the performances of the networks, obtained using different algorithms and protocols, some parameters need to be defined. In the context of WSNs the most significant parameters are coverage and lifetime. Coverage is known being the first QoS parameters because better is the coverage higher is the number of events detected reliably, while lifetime is directly related to energy consumption and represents the time that the network can work properly. In this section major performance parameters will be defined and discussed, while in subsection 1.4.1 will be summarized as, some schemes of controlled mobility known in literature, are able to improve some of these parameters.

1.3.1 Energy consumption and lifetime

Network lifetime has become the key characteristic for evaluating sensor networks in an application specific way. Especially the availability of nodes, the sensor coverage, and the connectivity and more in general the quality of service have been included in discussions on network lifetime. Network lifetime is perhaps the most important metric for the evaluation of sensor networks. Network lifetime as a measure for energy consumption represent an upper bound for the utility of the sensor network as the network can only fulfill its purpose as long as it is considered "alive", but not after that. In a scenario where deployment is not straightforward, if the metric is used in an analysis preceding a real-life deployment, the estimated network lifetime can also contribute to justifying the cost of the deployment. Network lifetime strongly depends on the lifetimes of the single nodes that constitute the network, thus, if the lifetimes of single nodes are not predicted accurately, it is possible that the derived network lifetime metric deviates in an uncontrollable manner. The major amount of energy is consumed by a sensor node during sensing, communication, and data processing, but among these activities the data transmission and reception is the activity that requires more energy [25]. Many researchers have focused on lifetime studies because the recharging or replacement of batteries is not feasible in many scenarios (too many nodes, hostile environment, etc.), and thus the lifetime of the network cannot be extended infinitely. For this reason, the design of algorithms and protocols that are able to save energy has become strategic. Naturally, lifetime was then discussed from different points

of view, which led to the development of various lifetime metrics. Depending on the energy consumers regarded in each metric and the specific application requirements considered, these metrics may lead to very different estimations of network lifetime. Some of the metric that we have found in literature are the followings [33]:

- *Network lifetime based on the number of alive nodes* - The definition found most frequently in the literature is n-of-n lifetime. In this definition, the network lifetime ends as soon as the first node fails.
- *Network lifetime based on sensor coverage* - the time that the region of interest is completely within the sensing range of at least one sensor node, i.e. the region is covered by at least one node.
- *Network lifetime based on connectivity* - number of successful data gathering trips [31]. In [32] this is further confined to the number of trips possible without any node running out of energy.

1.3.2 Coverage and connectivity

As already mentioned optimal resource management and assuring reliable QoS are two of the most fundamental requirements in ad hoc wireless sensor networks. Sensor deployment strategies play a very important role in providing better QoS, which relates to the issue of how well each point in the sensing field is covered (i.e. within the sensing radius). However, due to severe resource constraints and hostile environmental conditions, it is nontrivial to design an efficient deployment strategy that would minimize cost, reduce computation, minimize communication, and provide a high degree of area coverage, while at the same time maintaining a globally connected network [34]. The *degree of coverage* at a particular point in the sensing field can be related to the number of sensors whose sensing range cover that point. It has been observed and postulated that different applications would require different degrees of coverage in the sensing field. For example, a military surveillance application would need a high degree of coverage, because it would want a region to be monitored by multiple nodes simultaneously, such that even if some nodes cease to function, the security of the region will not be compromised, as other nodes will still continue to function, whereas some of the environmental monitoring applications, such as animal habitat monitoring or temperature monitoring inside a building, might require a low degree of coverage. On the other hand, some specific applications might need a framework, where the degree of coverage in a network can be dynamically configured. An example of this kind of application is intruder detection, where restricted regions are usually monitored with a moderate degree of coverage until the threat or act of intrusion is realized or takes place. At this point, the network will need to self-configure and increase the degree of coverage at possible threat locations. A network that has a high degree of coverage will clearly be more resilient to node failures. Thus, the coverage requirements vary across applications and should be kept in mind while developing new deployment strategies.

Many wireless sensor network applications require one to perform certain functions that can be measured in terms of area coverage. In these applications, it is necessary to define precise measures of efficient coverage that will impact overall system performance. Two main types of coverage have been defined by Gage [35] as follows:

- *Blanket coverage* to achieve a static arrangement of sensor nodes that maximizes the detection rate of targets appearing in the sensing field;
- *Sweep coverage* to move a number of sensor nodes across a sensing field, such that it addresses a specified balance between maximizing the detection rate and minimizing the number of missed detections per unit area.

In this thesis, we will focus on both blanket and sweep coverage. In first case the objective is to deploy sensor nodes in strategic ways such that an optimal area coverage is achieved according to the needs of the underlying applications. Here, it is worth mentioning that the problem of area coverage is related to the traditional art gallery problem (AGP) [36] in computational geometry. The AGP seeks to determine the minimum number of cameras that can be placed in a polygonal environment, such that every point in the environment is monitored. Similarly, the coverage problem basically deals with placing a minimum number of nodes, such that every point in the sensing field is optimally covered under the aforementioned resource constraints, presence of obstacles, noise and varying topography. In latter case the objective is not a deployment but to find a mobility patten for UAV nodes that is able to reach full coverage of a given area of interest in a minimum time.

Along with coverage, the notion of connectivity is equally important in wireless sensor networks. If a sensor network is modeled as a graph with sensor nodes as vertices and the communication link, if it exists, between any two nodes as an edge, then, by a connected network we mean that the underlying graph is connected, that is, between any two nodes there exists a single-hop or multi-hop communication path consisting of consecutive edges in the graph. Similar to the notion of degree of coverage, we shall also introduce the notion of *degree of network connectivity*. A sensor network is said to have k -connectivity if removal of any $(k - 1)$ nodes does not render the underlying communication graph disconnected. Like single degree of coverage, single-node connectivity is not sufficient for many sensor network applications because the failure of a single node would render the network disconnected. It should be noted that robustness and throughput of a sensor network are directly related to connectivity. Area coverage and connectivity in wireless sensor networks are not unrelated problems. Therefore, the goal of an optimal sensor deployment strategy is to have a globally connected network while optimizing coverage at the same time. By optimizing coverage, the deployment strategy would guarantee that optimum area in the sensing field is covered by sensors, as required by the underlying application. By ensuring that the network is connected, it is also ensured that the sensed information is transmitted to other nodes and possibly to a centralized base-station that can make

valuable decisions for the application. An interesting well-known result is that given a sensor network with degree of coverage k , if the transmission radius is at least twice than the sensing radius then the network is also k -connected [126]. This assumption regarding the relation between transmission and sensing radius will be often used in this thesis so that k -coverage guarantees also k -connection.

1.3.3 Throughput and delay

In a wireless ad hoc network with n nodes each of them equipped with an omnidirectional antenna, under a random network configuration, each node has a throughput capacity in the order of $\Theta(1/\sqrt{n} \log n)$ [37]. Even under an optimal arbitrary network configuration where the location of nodes and traffic pattern can be optimally controlled, the network could only offer a per-node throughput of $\Theta(1/\sqrt{n})$. The per-node throughput is decreased when the number of nodes increases. In fact, all the nodes in such network are sharing the same medium to transmit. When a node transmits, its neighboring nodes are prohibited from transmitting due to the interference. Therefore, the network throughput is interference-limited. As a consequence of [37] is that a small transmission range is necessary in order to limit the interference and thus improve the throughput. For this reason most of recent studies in WSNs assume a small transmission range for each sensor node, however, a smaller transmission range means that a packet needs to be transmitted through more hops, which inevitably leads to higher transmission delay [111]. In summary increasing the transmission radius can reduce the average number of hops and can reduce the transmission delay even if it will inevitably cause higher interference which leads to the lower throughput. Exist a lot of studies concentrate on optimizing the trade-off of the delay and the capacity. What we wanted point out in this brief subsection is the relation between transmission range in respect to number of hops, throughput and delay. Observations that need to be taken into account every time that some nodes parameters need to be tuned at priori or in a self-organizing way.

1.4 Mobility

Recently, wireless self-organizing networks are attracting a lot of interest in the research community. Moreover, in the last decade many mobile devices have appeared in the market. Exploiting mobility in a wireless environment, instead of considering it as a kind of disturbance, is a fundamental concept that the research community is beginning to appreciate now. Of course, the advantages obtainable through the use of the mobility imply the knowledge of the different types of mobility and the way to include it in the management architecture of the wireless networks [42].

We can identify three macro-categories of mobility: random, predictable and controlled. In the first category, mobile devices are supposed to move according to a random mobility pattern. Many probabilistic models have been proposed in order to foresee devices' movements. Unfortunately random mobility represents more of a problem to solve than an advantage to exploit. A network access point mounted on a means of public transportation that moves with a periodic schedule represents a case of predictable mobility. A predictable schedule permits an easier, programmable accomplishment of some desired target, but mobility is not considered as a network primitive yet. Finally, controlled mobility generally consists of mobile devices introduced in the network and moving to specified destinations with defined mobility patterns for specific objectives. We can figure out many goals that could be achieved through controlled mobility, such as: coverage management [9, 10, 11], energy consumption reduction [5, 6, 7, 8], transport layer parameters' improvement [16, 17, 18].

1.4.1 Advantages and Limitations of Controlled Mobility

Controlled mobility has been a hot research topic of the robotics community for many years. It concerns the motion coordination of a group of robots for a common objective, typically the coverage of a geographical area. But, the number of applications where controlled mobility is beneficial is enormous, and it spreads from underwater monitoring of seismic movements to planet exploration, from environmental sensing to site surveillance and localization of intruders. The coordination requires communication, computation and control among the robots. All these aspects are covered by the vast literature of theoretical and practical results in the control theory. Instead, in the networking research world, mobility has always been seen as an issue to face more than as a facility to exploit. Only recently, has controlled mobility gained an important role also for communications matters. In the two following subsections we intend to give an overview of the possible advantages, mostly taken from recent research works, and of the limitations and the choices that a network designer should consider in order to profitably use controlled mobility.

Advantages

As witnessed by the recent contributions in the wireless sensor, multihop, mesh and mobile ad hoc networking, controlled mobility offers several advantages to all those kinds of wireless networks which aim to an autonomous self-organization.

The first class of parameters which can be optimized by introducing controlled mobility in wireless networks is related with **power efficiency**. In [5] the authors present a distributed, self-adaptive scheme of mobility control for improving power efficiency while maintaining connectivity in a wireless sensor network. More important is that they introduce mobility as a network control

primitive.

Power consumption is also investigated in [6] and [7]. In [6] the authors discuss the usage of controllably mobile elements in a network infrastructure in order to reduce the energy consumption. They show that for increasing nodes densities, the presence of a mobile base station reduces the energy usage with respect to a network of static nodes. The mobility pattern of the mobile node is designed so that the path is fixed, but the speed profile followed along the path is flexible.

In [7] the authors split the nodes of a Mobile Ad-Hoc Network (MANET) in two categories: relay nodes, which are considered all mobile, and tracking nodes, which are static and used for getting information on a possible intruder of the network. They incrementally find the relays positions that minimize the total required transmission power for all the active flows in the MANET. A distributed annealing algorithm has been used for governing the motion of nodes.

A different approach is considered in [8], where only one node is considered mobile. It can be the sink or a relay node. For the case where the sink is mobile, the upper bound on the network lifetime is analytically determined to be four times that of the static network. The authors claim that a mobile sink is not feasible, because the sink is expected to be static since it acts as gateway to a backbone network. For this reason they assume the mobile node is a relay and they construct a joint mobility and routing algorithm in order to make the network lifetime come close to the upper bound.

An even more intuitive benefit comes from the ability to control the **coverage** and **connectivity** of the network, by modifying the positions of the nodes. A static network suffers from several disadvantages in covering a geographical area. First, even when the initial deployment did not leave regions uncovered, a static network can not cope with the dynamics of the environment and with the local disconnections. Second, the fixed positions of the nodes represent an easy target for a malicious attacker. In [87] the authors established necessary and sufficient conditions for the coverage to imply connectivity. In [85] authors investigated *k-coverage* and *k-connectivity*. *K-coverage* implies that every sensor node is linked with at least k sensors, and *k-connectivity* implies that for every pair of nodes there exist at least k different and disjoint paths that link these nodes. In [88] and [89] sensors are mobile nodes and are able to self-configure through some specific criteria that guarantee coverage of a region. Specifically, in [88], the authors implement an optimal placement strategy based on the concept of *potential field*. In [89] the authors develop a greedy algorithm based on controlled mobility. Controlled mobility can be efficiently exploited during the deployment phase, when the optimal displacement is too expensive or difficult to attain. In [18], the authors deform the topology of a multi-hop wireless network, by moving nodes in order to create new links. They show how this deformation reduces the average end-to-end delay and performs better than the approach where an increased capacity is assigned to the most congested data channel. The algorithm they

propose is centralized and needs the knowledge of the network topology. The algorithm changes the connections between the moving nodes at each step. At each step connectivity is preserved and the average delay decreases.

Furthermore, if a specific **nodes displacement** is shown to be optimal for some objective, controlled mobility is the way to achieve it.

In [9] the authors design adaptive and distributed algorithms, based on Voronoi diagrams, in order to coordinate a multi-vehicle network to meet on an event point following a predefined distribution.

Butler and Rus in [10] obtain the same objective making nodes cover a given area and converge on specific points of interest in a distributed fashion. The novelty is in the absence of placements defined a priori, and in the presence of new constraints, added so as not to leave any portion of the environment uncovered.

A more theoretical study is presented in [11], where the authors consider two metrics of quality of coverage (QoC) in mobile sensor networks: the fractions of events captured and the probability that an event is captured. They provide analytical results on how these two performance metrics scale with the number of mobile sensors, their velocity patterns, and event dynamics. They also develop an algorithm for planning sensor motion such that the probability that an event is lost is bounded from above. In our opinion, an algorithm based on this work, would need each sensor to be programmed accordingly with the mobility pattern computed by a centralized unit.

Controlled mobility can be effectively used during the network **deployment** phase, when an optimal placement of the nodes is too expensive or impossible due to environmental impedimenta. Reference [12] exploits the virtual force field concept for enhancing coverage. Sensors start from an initial random configuration, and, by using a combination of attractive and repulsive forces, they move to a final placement, where the area covered by each of them is maximized. The authors of [13] design two sets of distributed protocols, based on Voronoi diagrams, for controlling the movement of sensors to achieve target coverage. One set minimizes communications among the sensors, while the other minimizes movements. Coverage, deployment time, energy consumption and moving distances are the performance evaluation parameters used to show the effectiveness of their algorithms.

Controlled mobile sensors can also be used for **exploration and localization**, as in [14]. The cited work defines a hybrid architecture, made of a certain number of mobile actuators and a larger number of static sensors. The actuators move in the sensor-field and get information from the static sensors in order to perform site exploration, coverage repair and target localization. The algorithms which drive the actuators in their tasks are based on potential field and swarm intelligence.

Load balancing in wireless sensor networks is studied in [15]. The nodes closest to the base station are the bottleneck in the forwarding of data. A base station, which moves according to an arbitrary trajectory, continuously changes the closest nodes and solves the problem. The authors find the best

mobility pattern for the base station in order to ensure an even balancement of network load on the nodes.

It is well known that, in a wireless network, the **throughput** degrades with the number of hops. A node, which can act as a mobile relay, would limit the number of hops and increase network performance. In literature we can find many works on data mules, whose predictable mobility is also used for improving the **delivery ratio** of data. In delay-tolerant networks, Message Ferrying exploits controlled mobility in order to achieve the same task of transporting data with a high delivery ratio and also where end-to-end paths do not exist between nodes. In [16] the authors propose a scheme which manages with multiple ferries and is able to meet the traffic demands while minimizing the average data delivery delay.

A sensed phenomenon may require different rates of sampling by the sensor nodes, this leads to a non-uniform distribution of sensed data on the network and, without an accurate scheduling strategy of data collection, to a possible overflow of the buffers. In [17] the authors use mobile nodes for data gathering. First they show that the scheduling of multiple mobile elements with **no data loss** is a NP-complete problem, then they compare the performance of some computationally practical algorithms for single and multiple mobiles in terms of amount of overflow and latency in the collection of data.

In [18] the authors deform the topology of a multi-hop wireless network by moving the nodes to create new links. They show a reduction in the mean **end-to-end delay** of the network, even more effective than the alternative approach of increasing the capacities of the most congested network links. The algorithm is centralized and it takes as inputs: the network topology, the coordinates of the wireless nodes and the network load. Then it tries to change the network connectivity by moving the non-static network nodes in small steps. This is done such that, at each step, the network remains connected and its characteristic timescale goes down. In [20] authors propose an algorithm for joint relay node placement and node scheduling in wireless networks. They consider a system that consists of a relay node with controllable mobility and multiple nodes that communicate with each other via the relay node. The objective of their algorithm is to maximize the lowest weighted throughput among of all nodes. In [21] authors propose DARA, a Distributed Actor Recovery Algorithm. In particular they consider two versions of the algorithm to address different connectivity requirement. They claim DARA minimize the movement overhead imposed on the involved actors. Akkaya et al. DAPRA algorithm to detect possible partitions and restore network connectivity through controlled relocation of movable nodes [22]. In [23] authors propose two deployment algorithms to achieve sensor energy balancing and small amount of deployment energy consumption. They also consider mobility of nodes and exploit it.

In Table I all the cited works are shown along with the type of wireless network under investigation and the objective of the research. In order to offer a better categorization, in the three following columns, we put the number of

devices which are considered mobile, the scheme and the type of controlled mobility used in the algorithm.

Table 1.1. RELATED WORKS

Algorithm	Type of Network	Objective	# of Mobile Devices	Scheme	Type of Controlled Mobility
Mobility Control [5]	Multihop WN	Connectivity Energy Consumption	All	Distributed	Adaptive
Adaptive Motion [6]	WSN	Energy Consumption	Sink	-	Programmed
Distributed Annealing [7]	MANET	Energy Consumption	All	Distributed	Adaptive
ARALN [8]	WSN	Network Lifetime	Single node	-	Programmed
Coverage Behavior [9]	WSN	Coverage	All	Distributed	Adaptive
Local Voronoi [10]	WSN	Coverage	All	Distributed	Adaptive
BELP [11]	WSN	Quality of Coverage	To be computed	Centralized	Programmed
VFA [12]	WSN	Deployment	All	Distributed	Adaptive
VEC, VOR, Minimax [13]	WSN	Deployment	All	Distributed	Adaptive
TARANTULAS [14]	WSAN	Localization	Some	Distributed	Adaptive
Joint Mobility and Routing [15]	WSN	Load Balancing	Sink	-	Programmed
MURA [16]	DTN	Delivery Ratio	Some	Centralized	Programmed
MES [17]	WSN	No Data Loss	Some	Distributed	Adaptive
CD [18]	Multihop WN	End-to-end delay	All	Centralized	Adaptive
Joint relay node placement and node scheduling [20]	Wireless Networks	Throughput	Single Node	Centralized	Adaptive
DARA [21]	WSN	Connectivity	Some Nodes	Distributed	Adaptive
PADRA [22]	WSN	Connectivity	Some Nodes	Distributed	Adaptive
Deployment Algorithms [23]	WSN	Coverage and Connectivity	-	Distributed	Adaptive

Another work worth mentioning is [19]. It is not listed in Table I because the authors do not present an algorithm for the optimization of some parameters, instead they propose Morph as a new vision of sensor networking, where controlled mobility is considered as an additional design dimension of the communication protocols. They argue that, in Morph, controlled mobility can be employed for the sustainability of the network, which consists in both alleviating the lack of resources and improving the network performance.

Limitations

In spite of all the expected advantages that controlled mobility provides for self-organizing networks, it also poses many new questions which need to be answered before designing a protocol which envisions its employment.

First: Can the benefits introduced by controlled mobility counterbalance the expenses required by the **additional hardware**? A mobile node might need several extra devices for the motion, such as steering, positioning and navigation systems, in some cases it could be helpful to foresee also a remote control. Second: Does the **energy budget** allow the movements of one or many nodes of the network? It is necessary to define an energy model related with nodes' motion and one related with the communication needed for their coordination. For the former a simplified model is a distance proportional model

$E_m(d) = kd + \gamma$, where d is the distance to cover, $k[J/m]$ takes into account the kinetic friction, while $\gamma[J]$ represents the energy necessary to overcome the static friction, both these constants depend on the environment (harsh or smooth ground, air, surface or deep water). For the latter, usually the energy required to send one bit at the distance d is $E_c(d) = \beta d^\alpha$, where α is the exponent of the path loss ($2 \leq \alpha \leq 6$) depending on the environment and β is a constant [$J/(bits \cdot m^\alpha)$].

Third: If it is needed by the application, can the controlled mobility guarantee the **connectivity** at the intermediate steps in the process of new displacement of the nodes? Applications with real-time, delay-sensitive, continuous flow of data constraints do not tolerate node failures or local disconnections. For these kinds of applications, mobility should only operate when connectivity is not compromised by nodes movement.

Fourth: What **number of nodes** should be mobile? How is the **mobility pattern** chosen? Considering that only the sink is mobile brings evident advantages [6, 8, 15] but, usually, that specific node is in charge of bridging self-organizing and backbone networks. However, a reduced number of mobile nodes represents a viable answer to the power consumption problem and an effective solution for some parameter optimization [14, 16, 17]. A trade-off between the number of mobile nodes and the energy consumption has to be determined depending on the application demands. This compromise has to take into consideration also the determination of a mobility pattern that, if it is not selected appropriately, can cause a larger waste of resources.

Fifth: What happens if **network dynamics** are too fast with respect to nodes convergence through mobility? And if, for some reason, nodes can not reach the final, expected configuration, may an **intermediate displacement** be disadvantageous? A wrong placement of the nodes can transform the positive effects that controlled mobility was meant to introduce into a worse overall performance of the network. As seen in Table I, many schemes give the nodes the ability to move adaptatively to a task to perform or an event to track. This behavior is a primary concern in self-organizing networks design, since it gives the system the ability to react, in a distributed or a centralized fashion, to the changes in the applications demands and in the time-space constraints.

1.4.2 Challenges layer by layer

Fig. 1.1 (a) describes a self-organizing sensor network where, initially, nodes are monitoring their Voronoi cells. Reactively to specific events, as in Fig. 1.1 (b), nodes move in order to accomplish different objectives:

1. Establish a data flow between two points of the area, by moving according to a placement which guarantees the improvement of some figures of merit, such as throughput, end-to-end delay, power consumption, etc.
2. Surround, as quickly as possible, the place where an event happened, in order to gather all the required information on the phenomenon and monitor, as long as it is needed, its effects on the area under investigation.

3. Search for an intruder in the network, localize it, analyze its behavior and follow its movements.

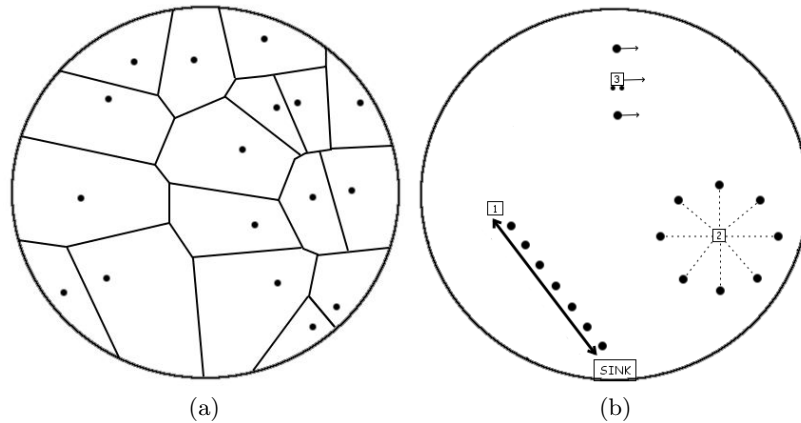


Fig. 1.1. Wireless sensors monitoring their Voronoi Cells (a), Wireless sensors accomplishing different tasks (b).

To attain such diverse goals, nodes move according to adaptive, distributed, task-aware motion coordination and communication protocols. In this section we try to focus on the research challenges issued by the incorporation of mobility control and task coordination in the protocol stack of a wireless self-organizing sensor network. Since the number of possible objectives that nodes are called to carry out in a self-organizing network can be large, but it is still limited, in our discussion, we decide that nodes can switch among different operational modalities. Each modality corresponds to a task that nodes are trying to achieve. The behavior of the protocol stack will depend on the current modality and the modality is selected and managed by the Mobility and Task Control Plane which is transversal to the protocol stack., as shown in Fig. 1.2.

Physical layer

If the physical layer could order nodes to move, it would be, first of all, to avoid or to limit wireless channel problems. In fact, the presence of environmental obstacles and multipath effects may cause the quality of connectivity to be vastly different in different regions of the network [6]. When, instead, a task has been assigned to a node, the physical layer can highly improve the performance by determining the most suited modulation and error control coding schemes or the nodes positions which allow the least energy consumption. Lately, the physical layer community is putting a lot of effort toward

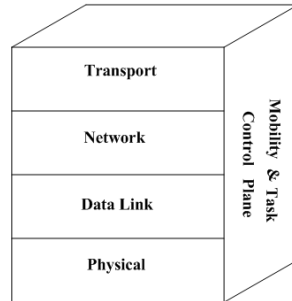


Fig. 1.2. Protocol stack including Mobility and Task Control Plane.

the direction of exploiting cooperation among nodes through recent research works which include Virtual MIMO and physical layer relaying.

Data Link Layer

The continuous exchange of information between the data link layer and the mobility control can be very advantageous for the network. By knowing positions, velocities and trajectories of the nodes, the data link layer can optimize the multiplexing of data streams and build accurate collision domains. In such a dynamic environment a fine time-space scheduling policy is needed in order to avoid wasteful retransmissions and prolong network lifetime by allowing a power-saving mode of operation. Also classical problems of this layer, such as the hidden terminal or the exposed terminal, should be newly investigated because through the usage of nodes motion they could be overcome.

Network Layer

Many procedures, typical of the network layer, are affected by the controlled mobility. Because of the dynamicity of the network, neighbour discovery becomes even more important, and it should be integrated with task discovery in order to facilitate the network organization. The network organization, in turn, is substantially modified by the movements of nodes and the assigned tasks. If sensing a phenomenon requires a clustered or a hierarchical scheme, establishing a data flow between two nodes of the network needs a linear topology. Routing has to be newly designed, because before it determined the most efficient path among deployed and static nodes, now it can look at the optimal positions where nodes have to move.

Transport Layer

Depending on the task, the transport layer issues several research challenges. In general, the end-to-end paradigm does not seem to be able to keep track

of dynamic multi-hop network characteristics in a responsive manner, so it should be replaced by hop-by-hop schemes, both for the rate and congestion controls. More specifically, the behavior of this layer cannot be dictated by the Round Trip Time (RTT), which can vary remarkably in different areas of the network or for different tasks. The scheduling and the fairness of data flows should take into account the availability of good quality links at programmable time instead of constant lower quality connections [19]. The measurements of the data to be transmitted, the data aggregation and the multiple flows relaying have to be reconsidered in order to meet application layer constraints.

Cross-layer Integration

Cross-layer integration is a “must” for controlled mobility in self-organizing networks. Different layers of the protocol stack will probably ask the nodes for diverse mobility patterns or final positions. For example, in Fig. 1.1 (b), when the assigned task is the establishment of a data flow between two points of the network, the network layer expects nodes to be arranged in a line in order to reduce energy consumption [5], instead, for the presence of an obstruction in the linear route, the physical layer is likely to demand a different optimal solution, even for the same objective. Likewise, the data link layer and the transport layer can have contrasting requirements in chasing an intruder - the former would try to isolate the latter, in order to have a more responsive information swap among the nodes involved in the pursuit. Generally, mobility and task control plane should constantly exchange data and requests with any layer of the stack, so as to represent the joint which makes a real cross-integration possible.

1.5 Self-organizing Systems (SOS)

Self-organization is a great concept for building scalable systems consisting of a large number of subsystems. Key factors in similar environments are coordination and collaboration of the subsystems for achieving a shared goal. Self-organization is not an invention but its principles have been evolved in nature and in the last few years, the concept of self-organization has been applied to technical systems and finally to wireless networks. Also, in this context the self-organization concept can be summarized as the interaction of multiple components on a common global objective. This collaborative work may be without any central control and the primary objectives of similar networks are scalability, reliability and availability [19]. Scalability is the ability of a system to handle growing amounts of work in a graceful manner or its ability to be enlarged to accommodate that growth, reliability is ability of a system to perform and maintain its functions in routine circumstances, as well as hostile or unexpected circumstances, finally availability regards the possibility to use the network whenever is needed consequently Self-Organization should not

require special maintenance periods. Scientific community is not completely agree about the definition of Self-Organization but the main characteristics can be summarized as follows:

Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of a system. The rules specifying interactions among the systems components are executed using only local information, without reference to the global pattern [26].

Moreover is important to distinguish between self-organized systems and systems that are only self-ordered, critical is the distinction between pattern and function, so in Self-Organization the global order that emerge through a formation or a specific pattern need to be functional to something [28]. Two more properties emerge from such definition:

- is not straightforward to guess the final pattern looking only to the local interactions (emergent behaviour);
- the missing of determinism of the algorithms [27].

For these features to hold, the following are some of the conditions that must be met [28]:

1. the system must have inputs and some measurable output;
2. the system must have a goal or goals;
3. the units must change internal state based on their inputs and the states of other units;
4. no single unit or non-communicative subset of units can achieve the system's goal as well as the collection can;
5. as it gains experience in a specified environment, the system achieves its goals more efficiently and/or accurately, on average.

One key research issue in designing and operating WSNs is to gain such self-* properties as:

- Self-configuration - allows WSN applications to configure their own operational parameters (e.g. routing decision parameters or sleep periods) depending on the current situation in terms of environmental circumstances, e.g. connectivity, quality of service parameters and self-organize into desirable structures and patterns (e.g. routing tables or duty cycling patterns);
- Self-management - capability to maintain devices or networks depending on the current parameters of the system;
- Self-optimization - allows WSN applications to constantly seek improvement in their performance by adapting to network dynamics with minimal human intervention;
- Self-healing - allows WSN applications to autonomously detect, localize and recover automatically from disruptions in the network (e.g. node or link failures).

The self-* properties are important in WSNs because they are often required to operate in unattended areas (e.g. forest or ocean), physically unreachable areas (e.g. inside a building wall) or potentially harsh/hostile areas (e.g. nuclear power plants).

The design of self-organizing systems is not top down as in traditional systems which are typically built starting considering the overall system and then approach the singles components and modules. Typically in design Self-Organizing systems the approach starts from thinking at the local interaction among components that, if they are modeled properly, could led to some kind of organization even if there is no guarantee about that [30].

Self-organization can be realized through different approaches [27]:

- *Location-based mechanisms* - Geographical positions or affiliation to a group of surrounding nodes such as clustering mechanisms, are used to reduce necessary state information to perform routing decisions or synchronizations. Usually, similar methods as known for global state operations can be employed in this context. Depending on the size of active clusters or the complexity to perform localization methods, such location-based mechanisms vary in communication and processing overhead.
- *Neighborhood information* - Further state reduction can be achieved by decreasing the size of previously mentioned clusters to a one-hop diameter. In this case, only neighborhood information is available to perform necessary decisions. Usually, hello messages are exchanged in regular time periods. This keeps the neighborhood information up-to-date and allows the exchange of performance measures such as the current load of a system. Such methodology is exploited in Chapter 3 through some algorithms and protocols of controlled mobility.
- *Probabilistic algorithms* - in some cases for examples if messages are very infrequently exchanged or in case of high mobility, pure probabilistic methods can lead to optimal results without any use of state information. Statistical measures can be used to describe the behavior of the overall system or the behavior of single components in terms of next action to perform. Obviously, no guarantee can be given that a desired goal will be reached. This approach will be used in Section 3.4 for analysis and description of mobility patterns for UAV networks.
- *Bio-inspired methods* - Biologically inspired methods build a category that is composed of neighborhood-depending operations very similar to behaviour of some species present in nature as ants or fishes and birds. All objectives are addressed by using positive and negative feedback often using a reinforcement learning. Different techniques that use this kind of approach for coverage problem will be explored in Chapter 4.

With miniaturization of computing elements we have seen many mobile devices appear in the market that can collaborate in an ad hoc fashion without requiring any previous infrastructure control consequently mobility has a large impact on the behaviour of ad hoc networks [29]. This latter con-

sideration allows us to consider the mobility as a fundamental aspect of the self-organizing networks.

1.6 Conclusion

In this chapter we have given several notions in order to define the context of this thesis. First of all we gave the definition of Wireless Sensor and Actor Network as a classical WSN that possess feature of taking decision and act accordingly. In the works introduced in the next chapters the "action" is accomplished using the mobility in order to improve the networks performances, consequently the nodes of our networks are mobile robots or UAV that are equipped not only with sensors and communication devices but also with mechanical devices that allow the movement. To this end the concept of mobility in communication networks is investigated showing some related works that use mobility for network issues. Finally the concept of Self-Organized System is defined, focusing on techniques of coordination, cooperation and learning process that allow the network to behave in a self-organized fashion.

Modelling and Solving Optimal Placement problems

2.1 Introduction

A wireless sensor network (WSN) consists of a large number of energy-constrained, low-cost and low-power sensor nodes. Each sensor node is a device, equipped with multiple on-board sensing elements, wireless transmitter-receiver modules, computational and power supply elements and it is characterized by limited computational and communication capabilities.

The WSNs are becoming increasingly popular for monitoring spatial phenomena. Indeed, they are deployed to an area of interest to collect data from the environment, process sensed data and take action accordingly.

Typical applications of the WSNs include environmental control such as fire fighting or marine ground erosion, but also sensors installation on bridges or buildings to monitor earthquake vibration patterns and various surveillance tasks such as intruder surveillance on premises.

In many real situations the designer of the network has little control over the exact deployment of the network and this produces the irregular placement of sensor nodes. Furthermore, even when the initial deployment is regular, as time progresses nodes will consume energy and stop working in a non-uniform fashion. Other factors (e.g. the edge effect) can also cause the energy distribution to be non-uniform. Therefore, not only irregular initial deployment but also non-uniform energy consumption has to be taken into serious consideration in order to allow better performance in terms of network lifetime [5],[6],[7],[8]. If we allow the nodes to use self-aware actuation in order to reorganize the available resources and form a new functional topology in the face of run-time dynamics, we will improve energy consumption and consequently the lifetime of the network. At the same time, mobility of nodes can help to achieve a higher resolution in collecting data from the sensor field. In this case the sensing advantages outweigh the cost of motion if appropriate design choices are made [17]. Mobility in wireless networks has been considered for many years as an issue to be hidden from higher layers of the protocol stack. In the recent past, the perspective of mobility has changed a lot and it became

an advantage to exploit [9] - [16]. In fact, in this work we consider and focus on mobile nodes by considering mobility as a fundamental aspect of the network. Actuation ability introduces a new design dimension in wireless sensors networks, allowing the network to adaptively reconfigure and repair itself in response to unpredictable run-time dynamics.

We focus on a multi-task, self-organizing WSN, that can carry out different functional roles. For this reason, the mobile nodes are also capable of moving towards positions, which are optimal for the specific task, that the network has to perform. Depending on the specific application at hand, a particular criterion should be chosen and optimized.

Different single-objective optimization models are presented to mathematically describe the optimal sensor displacement problem in a WSN. Each model tries to capture the main features of the specific situation under consideration and attempts to optimize the most significant objective for the considered scenario. The energy consumption and the travelled distance are the main measures used to evaluate the sensor placement, determined by a central computation unit.

The models we developed here can also help to move sensors towards new positions in order to spend the energy in a more uniform fashion and improve the total energy consumption.

The validity of the considered optimization models, in terms of solution quality and computational effort, is evaluated experimentally, by considering different scenarios. The solutions obtained by applying the centralized strategy are also compared with different distributed approaches. In particular, the evenly spaced [5], the bidirectional energy spaced [43] and four other innovative distributed heuristic strategies, proposed here, are considered in the experimental phase.

It is worth observing that the mathematical models considered in this section have been also presented during SENSORCOMM 2009 [44]. The main differences between this work and [44] are mainly related to the strategies used to heuristically solve (in a distributed environment) the sensor placement problem. In particular, in [44] only the evenly spaced algorithm, taken from the scientific literature [5], has been considered. In this section, we propose and test four innovative heuristic approaches, that allow us to obtain satisfactory results, outperforming the state-of-art distributed approaches (i.e., evenly spaced and bidirectional energy spaced strategies) considered for comparison.

The rest of the chapter is organized as follows: the next section gives a brief overview of the related works. Section 3 presents the main features of the problem under study and gives the description of the proposed optimization models. section 4 is devoted to the presentation of the defined distributed strategies. The computational results, that confirm the validity of the proposed mathematical models, are reported in section 5; finally, conclusions are drawn in section 6. This work is presented in [1].

2.2 State of art

Many contributions regarding the adoption of optimization-based approaches to sensor networks problems have been proposed in scientific literature. In [45], the authors proposed an interesting approach to the dispatch problem, that is how to determine from a set of mobile nodes a sub-set of sensor nodes to be moved to an area of interest with certain objective functions, such that the coverage and the connectivity properties are satisfied. They developed two solutions to the dispatch problem: a centralized one and a distributed one. The former is based on a previous placement and they converted the problem to the maximum-weight maximum-matching problem with the constraint that energy spent to move sensors has to be minimized or the constraint that the average remaining energy after sensors moved has to be maximized. In the distributed version constraints are the same of the centralized version, but sensors are allowed to independently determine their moving direction.

However, the proposed distributed approach achieves poor results, because the distances that nodes have to travel are not included in the optimization model and the algorithm is based on greedy choices. Furthermore, authors in [45] do not focus on the data flows but on the coverage and connectivity of the whole network.

In [46], the authors formulated a constrained multi-variable non-linear programming problem to determine both the locations of the nodes and the data transmission pattern. Constraints they considered are: maximization of the network lifetime and minimization of total cost. They studied a planar network where they applied results obtained through optimal strategies and performance bounds for linear networks. The authors do not consider the possibility to move the sensors, instead they think of replacing dead nodes, which is more expensive and more difficult. Furthermore, they assume all nodes have the same energy, which makes the approach suitable only for the initial deployment of the network.

In [47], the authors proposed a Multi-Objective Metric (MOM), taking into account 4 different metrics for base station placement in WSNs. First, they considered coverage as the ratio of sensor nodes which can communicate with a BS via either single-hop or multi-hop. Second, they introduced fault tolerance as the ratio of sensor nodes after the failure of base stations. Third, the energy consumption computed as the average distance between sensor nodes and their nearest BS. Finally, they introduced the metric of average delay as the standard deviation of the degree of base stations, that is a measure of network congestion. However, in [47], authors only consider base station optimal placement. In our work, we assume all the nodes are equipped with a mobility support and every node is able to move toward an “optimal” location.

In [48], the authors modelled the coverage problem as two sub-problems: floor-plan and placement. The former consisting of sub-dividing the service area into well-defined geometric cells and the placement problem is, in this case, to assign the sensor nodes into a set of cells. The next step consists of

solving a single optimization problem, where the objective function is to maximize the coverage of the service area while not exceeding the given budget. In their work, the authors only consider the coverage as objective. For this reason, their approach cannot be configured as multi-objective as we considered in our work. Moreover, they consider a first phase called floorplan, in which the exact position of sensors is known. This latter aspect can not be considered available in many realistic applications of wireless sensors networks.

In [49], the authors formulated the placement problem as a combinatorial optimization problem where the objective function is the minimization of the maximum distance error in a sensor field under certain constraints. In [49], the authors face the placement problem, so they did not take into account the mobility of nodes. In practice, they did not consider a dynamic “environment”, in the sense that objectives and tasks could change over time. As a result our approach is entirely innovative.

In [50], the authors investigated the problem of optimal sensor placement. They reformulated the problem such that the dimension of the non-linear problem NLP is independent of all decision variables. Moreover, they extended the sensor placement problem, based on static process conditions, to linear dynamic processes. An additional contribution of this work is the exact conversion of the general NLP into a convex program. All these results show how the sensor placement problem can be solved using a branch-and-bound search algorithm.

In [51], the authors introduced the concept of lifetime per unit cost, that is the lifetime divided by the number of sensors deployed. They analyzed both large and small networks and found that it is not an easy task to determine the optimal configuration, in terms of lifetime per unit cost. For this reason, they tried to determine the optimal number of sensors to be deployed and the best strategy to deploy them in order to maximize the lifetime per unit cost. To this aim they developed a two step procedure: in the first phase, a greedy strategy is applied to optimize the sensor placement; the second step is a numerical approximation to determine the optimal number of sensor nodes. On the contrary of our approach, in [51] the authors consider the possibility to introduce and, consequently, to use an “optimal” number of sensors. Instead, we base our work on a pre-existing configuration and we try to exploit nodes in the best possible way. Certainly, this matches real network conditions in the case of wireless sensors networks, where nodes cannot be added in an easy fashion and nodes cannot be easily recharged.

2.3 Problem Statement and Mathematical Formulation

In this section, we describe the main features of the problem under investigation and we present the mathematical models, developed to address the optimal sensor placement problem, under different scenarios.

In what follows, we consider an event-driven wireless sensor network and we assume that the number N of sensor nodes involved in the relaying are known.

In this respect, it is worth observing that the N value is determined by taking into account the density of the nodes in the field and the length of flow. In fact, from the density of nodes in the network we can compute the area that, on average, each node should be able to cover in order not to leave any uncovered region. In turn, from the coverage area it is possible to determine the maximum transmission radius (r in Table 2.1). This is an upper bound on the transmission radius, because the random deployment can make it unnecessary for the node to transmit at the computed maximum. By dividing the length of the flow for the maximum transmission radius we obtain the minimum number of sensors needed for establishing a data flow between the source-destination pair.

Node N is the sink node, whereas node 1 represents the source node.

It is worth noting that we focus only on a single source at each time because we suppose that all the nodes in a certain area that detect the same event will aggregate their sensed data and elect, with some criteria, one of them to be the only source, as is typical in wireless sensor networks. Details about data aggregation and the mechanism of election of the source node are out of the scope of this section.

It is assumed that each sensor is powered by a non-rechargeable battery and E_i , $i = 1, \dots, N$ represents the initial energy of sensor i .

In what follows, F_t denotes the flow time length and v_i , $i = 1, \dots, N$ is the initial position of node i , whereas \tilde{E}_i and \tilde{v}_i , $i = 1, \dots, N$ represent the residual energy and the position of the nodes in the new placement.

Let P_{rec} be the minimum required power for a bit to be correctly received in an area of one squared metre and k the spent energy for a space-unit movement of a sensor [5].

On the basis of the previous considerations, it is evident that for each sensor node i , the residual energy \tilde{E}_i can be determined as follows:

$$\tilde{E}_1 = E_1 - k \|\tilde{\mathbf{v}}_1 - \mathbf{v}_1\| - F_t \cdot P_{rec} \|\tilde{\mathbf{v}}_2 - \tilde{\mathbf{v}}_1\|^2 \quad (2.1)$$

$$\begin{aligned} \tilde{E}_i = E_i - k \|\tilde{\mathbf{v}}_i - \mathbf{v}_i\| - F_t \cdot P_{rec} \left[\|\tilde{\mathbf{v}}_{i+1} - \tilde{\mathbf{v}}_i\|^2 \right. \\ \left. + \|\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_{i-1}\|^2 \right], i = 2, \dots, N - 1 \end{aligned} \quad (2.2)$$

$$\tilde{E}_N = E_N - k \|\tilde{\mathbf{v}}_N - \mathbf{v}_N\| - F_t \cdot P_{rec} \|\tilde{\mathbf{v}}_N - \tilde{\mathbf{v}}_{N-1}\|^2 \quad (2.3)$$

In this set of constraints we are taking into consideration two different sources of energy consumption: movement and data transmission through the distance travelled by each node $\|\tilde{\mathbf{v}}_i - \mathbf{v}_i\|$ and the final distance between each couple of neighbour nodes $\|\tilde{\mathbf{v}}_{i+1} - \tilde{\mathbf{v}}_i\|^2$, respectively.

It is worth observing that the conditions (2.1)-(2.3) are non linear and represent the first set of constraints, shared by all the proposed models.

In addition, a sensor placement is feasible if for each node i , $i = 1, \dots, N$ the residual energy \tilde{E}_i is non-negative. Thus, the satisfaction of the conditions reported below should be also ensured:

$$\tilde{E}_i \geq 0, \quad i = 1, \dots, N. \quad (2.4)$$

As in [52] the lifetime of a sensor is simply defined as the time before the sensor runs out of battery and it is not usable for forwarding data anymore.

In what follows, we introduce five optimization models, in which different specific performance measures are taken into account. Indeed, for each model, the optimality criterion is chosen by considering the specific task the network has to carry out.

2.3.1 Model 1: Maximization of the total residual energy

The main aim of the first model is to find the sensor placement for which the total duration of the network is maximized. In other words, the objective is to maximize the total residual energy. It can be represented as follows:

$$\max \sum_{i=1}^N \tilde{E}_i$$

subject to constraints (2.1)-(2.4).

In most WSN deployments, there is only a finite source of energy. In fact, sensor nodes are usually battery-powered. Besides, the nature of such devices preclude battery replacement as a feasible solution, while many sensor network applications demand that the network must operate for a long period of time. It is challenging to use energy resources in the most efficient way. The model considered represents a possible general solution to this and it can be used in all the situations in which it is not economically and logistically convenient to replace dead nodes. Typical cases are represented by environmental applications. In particular, we cite air quality monitoring, water quality monitoring, fire detection, etc. [53], [54], [55].

2.3.2 Model 2: Maximization of the minimum residual energy

The second model has been developed with reference to the practical situation in which the sensor network is characterized by a low value of density. In these cases, it is required that each sensor lasts as long as possible, in order to not lose the coverage of its area. In static WSNs, the common solution for maintaining connectivity is to deploy redundant sensor nodes. When sensor nodes fail, redundant nodes can be used for repairing connectivity. However, in many cases it is difficult to ensure that redundant nodes are available for replacement, especially for a network in which sensors nodes are deployed

with a low value of density. In this case it could be useful to consider the second model proposed here. A typical example is represented by the scenario considered in [56], in which mobile nodes are used as data carriers and forward data between disconnected components of the network to the base station. From a mathematical point of view, the second proposed model takes the following form:

$$\max \min_{i=1}^N \tilde{E}_i$$

subject to constraints (2.1)-(2.4).

In particular, the main aim is to find the sensor placement, in order maximize the lifetime of the *critical* sensor, i.e., the sensor with the minimum value of residual energy.

2.3.3 Model 3: Maximization of the number of nodes with residual energy above a threshold

The third model can be viewed as a middle way between the two models introduced above. Indeed, the main aim is to find a sensor placement which maximizes the number of sensor nodes, whose residual energy is above a chosen threshold value E_{thr} . This model can be used in all the practical situations in which the network is sub-divided in clusters and cluster heads should be powerful devices because they act as a router for many slaves [57].

In order to give a mathematical representation of this model, it is necessary to introduce a set of binary variables. In particular, to each sensor i , $i = 1, \dots, N$, is associated a binary variable x_i with the following meaning: x_i is equal to 1 if the residual energy \tilde{E}_i of i is greater than the threshold E_{thr} and is equal to 0 otherwise.

Model 3 takes the following form:

$$\max \sum_{i=1}^N x_i$$

subject to

$$\begin{aligned} x_i &\leq \frac{\tilde{E}_i}{E_{thr}}, & i = 1, \dots, N \\ x_i &\in \{0, 1\}, & i = 1, \dots, N \end{aligned}$$

and constraints (2.1)-(2.4).

2.3.4 Model 4: Minimization of the total travelled distance

Till this moment, we have considered the travelled distance only as a source of energy consumption (section 2.3), without giving to this parameter the importance it deserves for several reasons: first, in some practical applications,

a high mobility level could be dangerous or difficult to achieve because of the environmental conditions nodes have to work in; second, a longer travelled distance means not only a higher consumed energy, but also a higher usage of the electro-mechanical system in place for the movement; third, sub-optimal placement would ensure good results in terms of nodes' lifetime without making nodes move too much. At the same time, we already showed in section 2.1 how important and beneficial mobility is for the network, not only in relation to the consumed energy. Hence, the term $\|\tilde{v}_i - v_i\|$ has a central role in determining the tradeoff between the exploitation of mobility advantages and the constraints in its usage. Therefore, the last two models focus on the travelled distance as the optimization objective.

Typical examples of the aforementioned dangerous or environmentally difficult situations arise in the military field. In fact, WSNs can be used for battlefield surveillance and position tracking of the enemy [58], [59] and [60]. Whenever the total distance travelled by all the sensors to get to the final positions has to be minimized, we can use the mathematical representation of model 4, which is given in what follows:

$$\min \sum_{i=1}^N \|\tilde{v}_i - v_i\|$$

subject to constraints (2.1)-(2.4).

2.3.5 Model 5: Minimization of the maximum travelled distance

The distance travelled by the sensors, considered in model 4, is also taken into account in model 5 as performance measure. In this case, the main aim is to find the sensor placement for which the distance travelled by the sensor that moves the most is minimized. Similar to the previous model, in military applications it could be needed to move few nodes as little as possible.

Model 5 takes the following form:

$$\min \max_{i=1}^N \|\tilde{v}_i - v_i\|$$

subject to constraints (2.1)-(2.4).

The presented models can be used for different tasks and applications. Depending on the sensed data, the central solver determines the appropriate task/application, it maps it into the most suitable model and computes the solution as a new optimal placement.

2.3.6 Centralized Placement Algorithm

In this section we introduce the steps performed by an algorithm that allows the computation of the placement that is optimal for some specific objective.

In order to specify the behaviour of the algorithm, we need to make some general assumptions:

- sensor field is plane and free of obstacles;
- nodes know their own positions;
- nodes monitor their own residual energy;
- nodes are not equipped with directional antennae, so their transmissions are always considered omnidirectional;
- nodes are provided with the same mechanical devices, so they all move at the same velocity;
- residual energies are considered different for all the nodes;

Throughout this work this algorithm has been considered centralized, because it requires a central computation unit for:

- collecting the information from the nodes, specifically, residual energies, current positions and characteristics of sensed events,
- selecting the particular model depending on the objective,
- solving the model and determining the optimal placement,
- communicating the new placement to the nodes.

The algorithm to achieve the best placement uses the following steps:

1. when a new event is sensed by one or more nodes of the sensor field, the computation centre is informed about the most important characteristics of the event;
2. depending on the event's characteristics, the computation centre determines the objective and, consequently, it chooses the model to solve;
3. everytime a new event has been sensed, a bidirectional data flow between two nodes is activated¹;
4. everytime a new flow has been activated, a routing protocol occurs, establishing a path of mobile nodes between two static communicating nodes. We give more details about the used routing protocol in the next subsection;
5. when the data path is established, nodes report their current position and residual energy status to the central computation unit, so to allow it to store and use this information;
6. once the central computation unit has collected the information on all the nodes involved in the data flow, it solves the chosen model;
7. the result from the model is a new nodes placement that is forwarded to the nodes;
8. all the nodes that are ordered to reach a new position move towards it;
9. the new placement is achieved when the node that has to travel further reaches its optimal position. The time needed for this operation can be calculated in advance by the computation centre. Once this time has passed, the source is triggered to start the transmission.

¹ for sake of simplicity we assume that all the possible events trigger the activation of a bidirectional data flow between a source and a destination node.

In the performance evaluation section we will show the results obtained by this algorithm for different targets in comparison with a distributed one.

2.3.7 A routing technique

In this subsection we give some details about the used routing technique for selecting the sensor nodes in the establishment of a path between the terminal nodes. It is important to point out that the determination of the best routing scheme is out of the scope of this section which focuses on the optimization of the path-lifetime and the minimization of the energy spent in a data flow by exploiting nodes mobility. Thus, our choice of a routing procedure is only motivated by a fair comparison of performance, for this reason we considered a routing technique based on the results of [5]. For the same reason, in this work, we do not consider all the costs related to the information exchange for the routing. Indeed, the routing scheme selects the nodes which are close to the straight line between the two terminal nodes, and specifically, it selects the closest nodes to the evenly spaced positions. In order to better understand this mechanism we can observe the Figure 2.1, in which the full circles along the straight line represent the optimal solution as obtained in [5]. It is useful to outline that [5] did not consider residual energies of sensor nodes to compute the optimal solution. The marked empty circles, in Figure 2.1, are the nodes selected by the routing technique. These nodes are the nearest to the evenly spaced positions along the straight line between the terminal nodes. In our case the routing selection is realized by a central computation unit, but there are many distributed routing protocols for Ad Hoc or Wireless Sensor Networks that can be adapted for this matter [41].

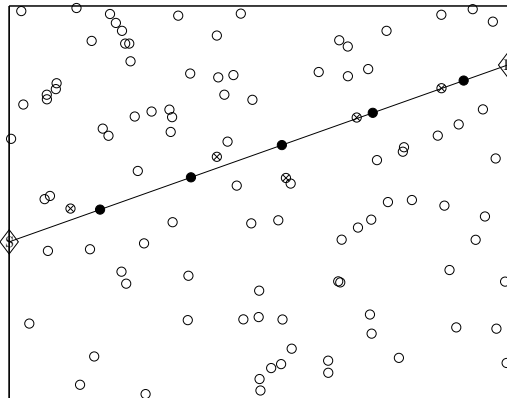


Fig. 2.1. An example of routing selection.

2.4 Distributed Heuristic Strategies

The centralized approach described in the previous section, which represents a sort of “exact” solution to the sensor placement, cannot be always implementable in a real-life setting. Moreover, computational times and communication issues could reduce the effectiveness of proposed solutions for each problem. Here we propose a set of distributed heuristic strategies which can be implemented locally for each node or group of nodes. They are based on the common evenly-spaced approach, even if other approaches with a minimal information communication among neighbor nodes can be adopted as well, like for example energy-spaced approach. These strategies could be implemented separately or, similarly to the previous approach, could be selected according to the specific application.

In order to illustrate these strategies we consider a simple network with just six nodes as depicted in Figure 1. Node 1 is the source node and 6 the sink node.

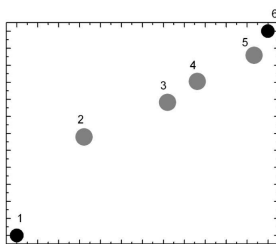


Fig. 2.2. Example sensor network

We suppose that at each node the position of its neighbors is known and that the process at each node can be asynchronous. The following schemes and figures describe the heuristic strategies we propose. In all the figures, white circles represent starting positions and light grey circles final positions, that is before and after each step of the procedures. At each step dark grey circles are not interested by the movement.

Procedure A - For each node i , $i = 1, \dots, 4$, define position of node $i + 1$ without moving node i (Figure 2).

- Process node 1: define position of node 2 without moving node 1.
- Process node 2: define position of node 3 without moving node 2.
- Process node 3: define position of node 4 without moving node 3.
- Process node 4: define position of node 5 without moving node 4.

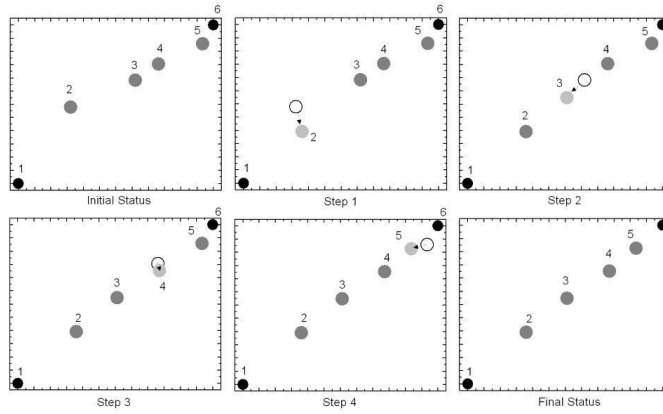


Fig. 2.3. Graphical Representation of Procedure A

Procedure B - For each node i , $i = 2, \dots, 4$, define position of nodes i and $i + 1$ (Figure 3).

- Process node 2: define position of nodes 2 and 3.
- Process node 3: define position of nodes 3 and 4.
- Process node 4: define position of nodes 4 and 5.

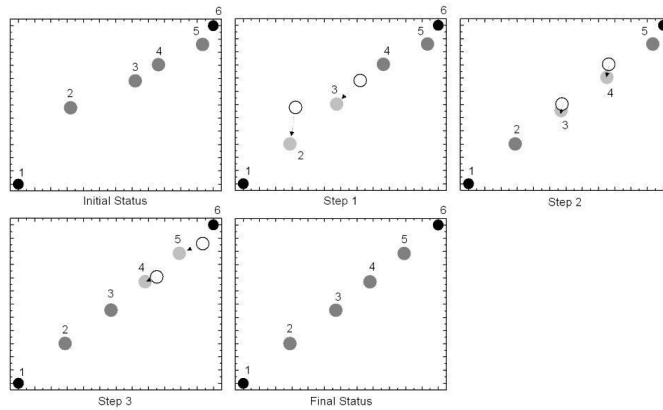


Fig. 2.4. Graphical Representation of Procedure B

Procedure C - For each node i , $i = 2, \dots, 5$, define position of nodes i (Figure 4).

- Process node 2 and 4: define position of nodes 2 and 4, without moving nodes 3 and 5.
- Process node 3 and 5: define position of nodes 3 and 5, without moving nodes 2 and 4.

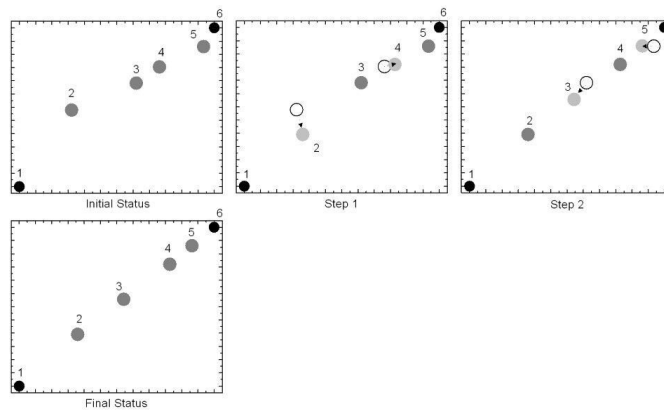


Fig. 2.5. Graphical Representation of Procedure C

Procedure D - For each node i , $i = 2, 4$, define position of nodes i and $i + 1$ (Figure 5).

- Process node 2: define position of nodes 2 and 3.
- Process node 4: define position of nodes 4 and 5.

The proposed heuristic strategies are easy to implement, because each sensor node needs just the information about its neighbors position and are very efficient since for every procedure each step requires only the execution of an arithmetic operation.

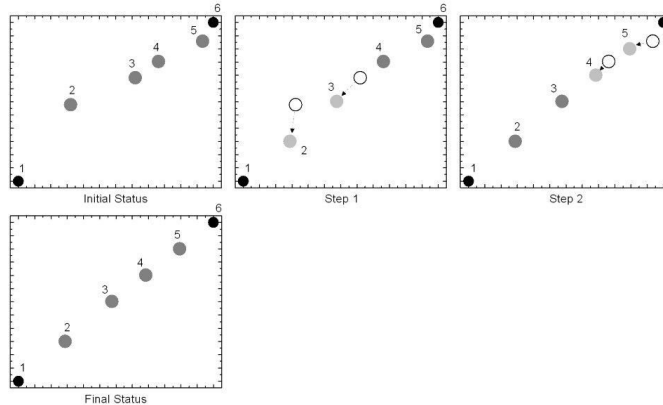


Fig. 2.6. Graphical Representation of Procedure D

2.5 Computational experience

In order to validate the effectiveness of proposed models and heuristics strategies a set of computational experiments has been carried out. In particular, the aim of this activity has been twofold. First of all, we have tested the significance and the performance of the optimization models, outlining how a centralized approach could improve the overall network performance. Secondly, we have analyzed the behavior of the proposed distributed heuristic strategies for each applicative problem considered w.r.t. the exact optimization models.

The models have been formulated and solved using LINGO², an integrated system with an algebraic modelling language and several linear and nonlinear solution kernels.

In order to evaluate the performance of our techniques we implemented the algorithms and the reference scenario in Matlab. We have considered as a test case a field area of $1000\ m \times 1000\ m$, where mobile sensors are deployed in a uniform random fashion and a bidirectional flow has been activated between a couple of nodes located in two opposite corners of the field. Relay nodes start with different residual energies in the range $15\text{-}20\ J$ and they are characterized by a maximum transmission radius r , a transmission rate r_T and a minimum required power for a bit to be correctly received P_{rec} .

The energy expenditure needed for the movement of the sensors has been considered, according to a simple distance proportional cost model: $E_M = kd$, where d is the travelled distance and k is a movement constant. The energy model used for the transmission is taken from [61]. We have considered five different values for the node density (200, 400, 600, 800 and 1000 sensors in

² www.lindo.com

the network). Table 2.1 summarizes all the parameters set in the simulation environment.

Table 2.1. Evaluation Parameters

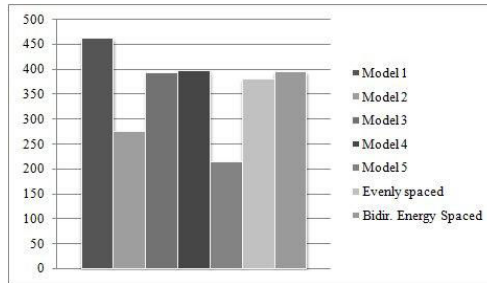
Field Area (LxL)	1000m x 1000m
Nodes Density (ρ)	$[2 \div 10] \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$
Source position (\mathbf{v}_1)	(0,0)
Destination position (\mathbf{v}_N)	(1000,1000)
Flow Time Length (T_F)	$87.6 \cdot 10^3 h$
Flow Length (l)	$1000\sqrt{2}m$
Max Transmission Radius (r)	$1/(2\sqrt{\rho}) m$
Relay nodes (N)	l/r
Initial Residual Energy Range (E_i)	$15 \div 20 J$
Min Req Power (P_{rec})	$3.16 \cdot 10^{-12} W/m^2$
Transmission Rate (r_T)	$1 kb/s$
Movement Constant (k)	$0.1 J/m$
Energy Threshold (E_T)	$10 J$
Runs for each scenario	100
Confidence Interval	95%

We have considered several instances of the test problem obtained considering different randomly generated values for initial positions and energies. The following figure (i.e., Figure 6) summarizes the results in terms of the criteria mean values measured on the considered test cases for each one of the proposed optimization models, together with those of two placement strategies, the evenly spaced one and the bidirectional energy spaced. The first one is a commonly adopted strategy, which however is effective just under specific conditions [5]. The second one is due to a recent contribution of Natalizio et al. [43], which shows how this placement strategy is optimal for sensors which are located along the flow direction.

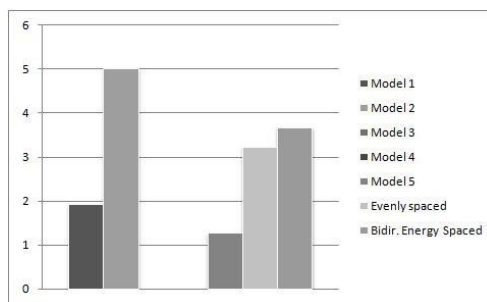
As we can see, each specially tailored optimization model outperforms the other ones and the two benchmark strategies w.r.t. the criterium it has been designed for. Table 2.2 reports the percentage worsening of the placements proposed by the optimization models and the benchmark strategies w.r.t. the optimal placement for each one of the considered criterium.

Table 2.2. Models comparison (% worsening)

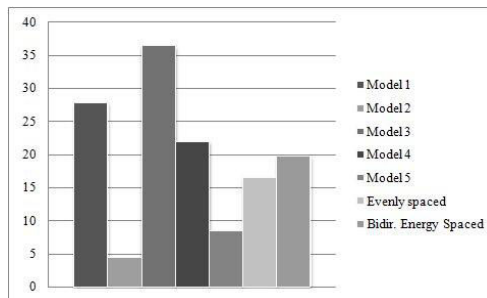
	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5
Model 1	0	61.51	23.69	40.81	52.39
Model 2	40.19	0	87.78	152.61	49.29
Model 3	15.13	100	0	88.46	63.28
Model 4	14.19	100	39.65	0	36.41
Model 5	53.38	74.8	76.56	158.94	0
Evenly spaced	17.63	53.69	54.61	154.2	57.41
Bidir. energy spaced	14.66	26.72	45.69	125.56	56.34



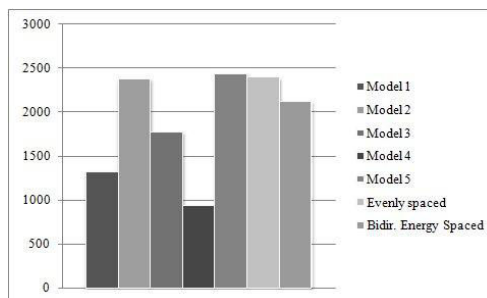
(a) Mean Overall Residual Energy



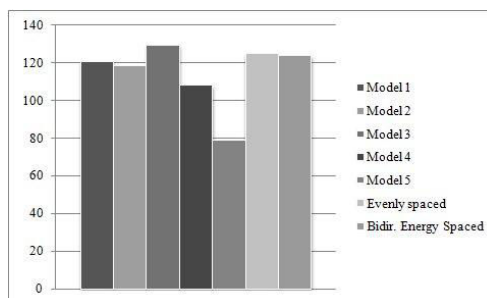
(b) Mean Minimum Residual Energy



(c) Mean number of nodes over the threshold



(d) Mean Overall Travelled Distance



(e) Mean Maximum Distance

Fig. 2.7. Computational Results obtained with the proposed Optimization Models

Even if the benchmark strategies could seem to be an acceptable choice, in particular due to their easy implementation, it is worthwhile noting that in some cases the placement they provide can be infeasible, that is some nodes do not have a sufficient energy level to guarantee the movement and the flow transmission.

We have also analysed the performance of proposed heuristic strategies by means of their comparison with each specially tailored model, which we call Best Optimization Model (BOM), and the benchmark strategies aforementioned. Figure 7 shows the results for the five criteria.

Similarly to the previous one, table 2.3 reports percentual worsening of each heuristic strategy w.r.t. BOM and benchmark policies.

Table 2.3. Heuristics comparison (% worsening from BOM)

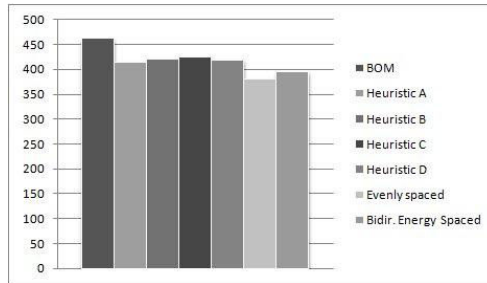
	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5	Avg.
Heuristic A	10.26	21.86	41.71	118.91	35.96	45.74
Heuristic B	8.85	35.99	38.08	114.18	43.5	48.12
Heuristic C	8.34	22.57	40.04	111.23	35.37	43.51
Heuristic D	9.3	26.57	40.43	116.16	37.64	46.02
Evenly spaced	17.63	35.58	54.61	154.2	57.41	63.89
Bidir. energy spaced	14.66	26.72	45.69	125.56	56.34	53.79

As we can see, the heuristics performances are quite similar and are almost always better than those of benchmark strategies.

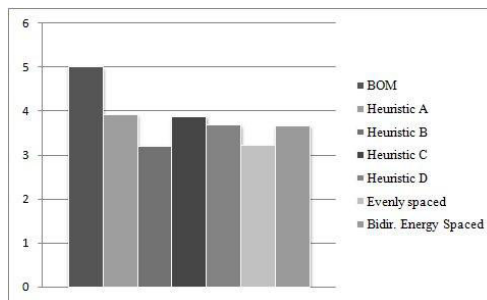
In particular, except for objective 4, the performance worsening w.r.t. optimization models is acceptable, varying from 8% to about 40%. Analysing the performance of the proposed strategies, we can observe that strategy C can be considered as the best one, because it outperforms the other ones in almost all the cases. The column of table 2.3 confirms that strategy C is globally preferable to the other ones.

Moreover, the comparison with the performances of the aforementioned benchmark strategies shows that the proposed strategies are more effective for all the considered criteria, allowing significantly minor worsening levels. For example, as regards the first objective the worsening of benchmark strategies is twice that of the best heuristic strategy.

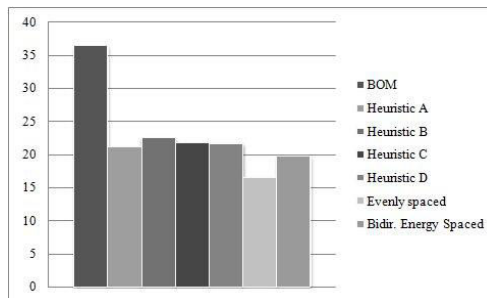
All these results, together with the difficulty concerning the implementation of centralized optimization-based policies, show how the proposed heuristic strategies could be a good approach in a real-life setting. From an applicative standpoint, since the placements suggested by the proposed heuristic strategies are similar in terms of the objective criteria, there is not a real advantage in switching from a strategy to another one. A good trade-off could be the implementation of just one (local) strategy, chosen according to its overall performance or to the most frequent applicative problems the network is expected to face.



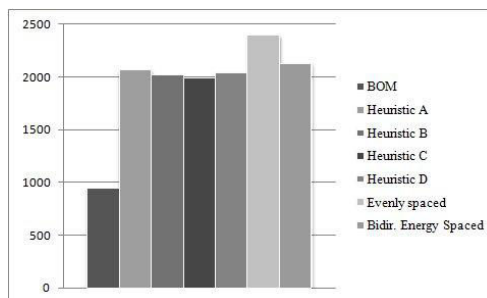
(a) Mean Overall Residual Energy



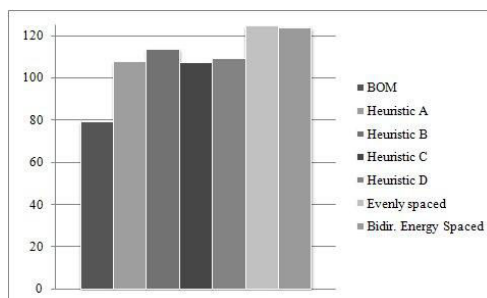
(b) Mean Minimum Residual Energy



(c) Mean number of nodes over the threshold



(d) Mean Overall Travelled Distance



(e) Mean Maximum Distance

Fig. 2.8. Computational Results obtained with the proposed Heuristic Approaches

2.6 Conclusions

In this chapter, the optimal sensor placement in wireless sensor networks has been addressed. Some optimization models, in which different criteria are taken into account, have been considered. The considered models, implemented through a centralized scheme, allow us to determine the most suitable sensors placement, depending on the specific task the network has to execute.

The centralized scheme has been compared with the performance achieved by a set of state-of-art and innovative distributed strategies, that require only local information.

The computational results obtained by applying the proposed distributed schemes are very encouraging. Indeed, the sensor displacements obtained with the distributed heuristics are comparable with those determined by the centralized approach. In addition, the proposed approaches outperform the state-of-art distributed strategies.

Controlled Mobility: algorithms and protocols

3.1 Introduction

In this chapter we will propose some distributed algorithms that, by using controlled mobility, allow each node to autonomously move towards a position that is optimal in terms of energy consumption during for the transmission of a data flow. The classical assumption in literature to optimize the energy consumption is to place nodes along the straight line between source and destination in the "evenly spaced" positions. This is the best placement when the initial residual energy is the same for all nodes, obviously this assumption is not realistic, so we want to remove it and determine the new placement for relay nodes based on the energy level (we refer to this scheme as "energy spaced" algorithm). A further improvement is proposed by making nodes move towards the final optimal placement introducing a mechanism named "virtualization of movements" that is able to significantly reduce the energy consumed for the movement.

A similar algorithm of mobility is also put to work along with a routing protocol that is "mobility aware" and consequently it is able to choose the relay nodes for a multi-hop communication. The chosen nodes are the closest to the final optimal position in order to minimize the travelled distances, taking into account also the residual energies of the nodes.

These algorithms show as mobility can impact the performance of the network so, finally, an analytical tool is proposed to measure the performance of different patterns of mobility in terms of coverage that is the main QoS parameter for such kind of networks.

In next section we show the existence of placements that are optimal in terms of energy consumption for data transmission. In section 3.3 a routing protocol using controlled mobility will be introduced, finally in section 3.4 an analytical tool for measuring the effectiveness of different mobility patterns in terms of coverage will be presented.

3.2 Optimal placements and Mobility Schemes for Improving Energy Efficiency

3.2.1 State of art

Wireless sensor networks are made up of a large number of sensors deployed in the area to be monitored. Due to their weight and size limitations, energy conservation is the most critical issue. Usually, the sensor nodes are deployed randomly and after the deployment, these nodes are generally stationary and self-organized into networks. Energy saving in a wireless sensor network can be achieved by deploying sensor devices in an ad hoc manner based on some specific task. Self-organizing is a fundamental concept to build scalable and dynamic systems. In similar systems, the key factors are coordination and cooperation among system's agents, in order to achieve a common task. In the last few years, the concept of mobility has been strictly related to sensor networks. A concept not deeply investigated for telecommunications applications is controlled mobility. This last concept is logically connected to design specific mobility control algorithms that usually need to be distributed. The design of mobility control algorithms is challenging, because different issues need to be addressed. Some of these issues have been addressed in [5]. Specifically, a totally distributed scheme, which enables mobility control in an asynchronous fashion, is developed. In order to prove the effectiveness of controlled mobility, the authors considered different application scenarios. However, the usage of nodes mobility is a new source of energy consumption. In this section we focus on the development of a new strategy, which takes advantage of wireless devices controlled mobility without this side effect. Specifically, a new concept of nodes movement is introduced: virtual movement. At each iteration, nodes compute their new position depending on the task they have to achieve, but they do not move immediately. Instead, the calculated position is considered "virtual". The algorithm converges when nodes are not able to find a different position in respect of the last one computed, and they physically move towards this position. The new concept of "virtual" movement of sensor nodes allows us to exploit the advantages of controlled mobility without wasting energy related to the movement.

For many years controlled mobility has been deeply investigated by the robotic research community, where the movement coordination of a group of robots was considered in order to achieve a common task. Several applications exist where the advantages of controlled mobility could be exploited. Traditionally the mobility of devices in a telecommunication network was considered a problem to overcome but in the last few years controlled mobility has become a new network design dimension. Controlled mobility can be considered in order to optimize different parameters in wireless sensor networks such as lifetime. In [5] the authors propose a controlled mobility algorithm that is distributed and adaptive. Their algorithm achieves better control of power management and specifically, it pursues better energy consumption in a

wireless sensor network, where nodes are always connected during their movement. The key factor of this work is that authors use controlled mobility as a new design primitive of the sensor network. In [6] and [7] energy consumption has been studied. In [6], the authors aim to obtain improvements in terms of energy consumption by exploiting controlled mobility. They show that, by using a mobile station, if the node density increases, the energy consumption decreases. They considered a mobility pattern where the path is previously known and speed varies. In [7] the authors consider two types of nodes: relay nodes, that are mobile nodes and tracking nodes, that are static nodes. Tracking nodes give information about network intruders. Mobile relays are used to find optimal positions while minimizing power transmission for each active flow. Nodes movement is driven by a distributed algorithm. In [8] the authors propose a different approach, in which only one node is mobile. This mobile node can be a sink or simply a relay node. Through an analytical approach, they show an upper limit for the network lifetime, when the mobile node is the sink. In this case, lifetime is four times larger than in the case where all the nodes are static. However, since the sink is usually a gateway and it should be static, they consider a relay node as mobile and develop a routing algorithm, whose performance is close to the upper limit.

Another interesting field of research related to the controlled mobility is the network coverage-connectivity area. In [87] the authors established necessary and sufficient conditions for the coverage to imply connectivity. In [85] authors investigated *k-coverage* and *k-connectivity*. *K-coverage* implies that every sensor node is linked with at least k sensors, and *k-connectivity* implies that for every pair of nodes there exist at least k different and disjoint paths that link these nodes. In [88] and [89] sensors are mobile nodes and are able to self-configure through some specific criteria that guarantee coverage of a region. Specifically, in [88], the authors implement an optimal placement strategy based on the concept of *potential field*. In [89] the authors develop a greedy algorithm based on controlled mobility. Controlled mobility can be efficiently exploited during the deployment phase, when the optimal displacement is too expensive or difficult to attain. In [12] Zou and Chakrabarty exploit the concept of virtual force to improve the coverage of a geographical area. At the beginning, nodes are placed randomly, each node is subject to attractive and repulsive forces from the other nodes, so that they all move towards a final displacement, where the area covered is maximized. In [90], the authors develop a distributed algorithm based on the concept of *artificial potential fields*, where the objective is to maximize the area covered and let every node have at least four sensors in its own transmission range. Controlled mobility can be exploited to reduce end-to-end delay. In [18], the authors deform the topology of a multi-hop wireless network, by moving nodes in order to create new links. They show how this deformation reduces the average end-to-end delay and performs better than the approach where an increased capacity is assigned to the most congested data channel. The algorithm they propose is centralized and needs the knowledge of the network topology. The algorithm

changes the connections between the moving nodes at each step. At each step connectivity is preserved and the average delay decreases. In [14] exploration and localization are obtained through controlled mobility. Authors introduce an hybrid architecture where a certain number of nodes is mobile and the others are static. Mobile nodes move inside a certain area and pick up information from static nodes to explore the area. The algorithm that makes nodes move is based on potential fields and swarm intelligence.

Another interesting field of research regards the optimal placement of mobile nodes for some specific target. Specifically, [5] is the first work where controlled mobility is used to place nodes in an evenly spaced manner on the straight line between the terminal nodes of a flow. In [24], the final positions of the nodes are not evenly spaced, because their initial energy level is considered in the positions calculation.

3.2.2 Optimal placements

In this section we report the mathematical framework introduced in [24, 43] for the determination of the optimal nodes placement in a Wireless Sensor Network. The optimality of the solution is meant in terms of energy consumption for a monodirectional or a bidirectional data flow. One of the novelty of our approach in [24, 43] is the consideration of different levels of residual energy for the nodes involved in a data flow between a source and a destination node. This assumption paves the way to the next step of our work, which consists in moving the nodes to the optimal positions by dynamically reacting to some external or internal trigger. Thus, this section presents the preliminaries which will be used by the real and virtual movement algorithms presented in the following sections, and it considers the monodirectional and the bidirectional data flow cases separately.

Monodirectional data flow

In [24] we investigated the problem of placing sensors involved in a monodirectional data flow in order to maximize their lifetime, when their residual energies are different.

The energy model we used to characterize the physical layer of our mathematical scheme is taken from [91]. By simplifying this model we obtain that the energy required to send one bit at distance d is $E = 2E_{elec} + \beta d^\alpha$, where E_{elec} [J/bit] is a distance independent term that takes into account the energy needed by the transceiver circuitry to transmit or receive one bit, α is the exponent of the path loss ($2 \leq \alpha \leq 5$) and β is a constant [J/(bit·m $^\alpha$)]. E_{elec} is counted twice, because it is assumed that $E_{elec} = E_{elec}^{trans} = E_{elec}^{rec}$ as in [91]. We set α equal to 2 and β equal to 10pJ/(bit·m 2), which are typical values of the free space model.

Next, we introduce the mathematical model of the system. Let \mathbf{v}_1 and \mathbf{v}_n denote the known source and destination positions, respectively. Let $\{\mathbf{v}_i\}_{i=2}^{n-1}$

be the positions of the $n - 2$ relay nodes. Let $\{T_i\}_{i=1}^{n-1}$ and $\{E_i\}_{i=1}^{n-1}$ be the life times and the residual energies of the nodes, respectively. Let P_{rec} denote the minimum required power in order for a bit to be correctly received in an area of one squared metre. We assume a power control system is in place so that the transmitter adjusts its power in order to deliver P_{rec} at the receiver. This implies that each T_i is a function of the positions of nodes i and $i + 1$, i.e. $T_i = \frac{E_i}{P_{rec} \|\mathbf{v}_i - \mathbf{v}_{i+1}\|^2}$. The distance between two successive nodes in the path is $\|\mathbf{v}_i - \mathbf{v}_{i+1}\|$.

Problem: Find $\{\mathbf{v}_i\}_{i=2}^{n-1}$ such that $\min\{T_i\}_{i=1}^{n-1}$ is maximized.

This can be immediately solved by placing the nodes on the segment with the extremes \mathbf{v}_1 and \mathbf{v}_n , the distance between adjacent nodes being chosen in order to have $T_1 = T_2 = \dots = T_{n-1} = T_{PL}$, where T_{PL} is the path-lifetime. This gives

$$\begin{aligned} \mathbf{v}_i &= \mathbf{v}_{i-1} + \sqrt{\frac{E_{i-1}}{P_{rec} T_{PL}}} \mathbf{u} \\ &= \mathbf{v}_1 + \sum_{k=1}^{i-1} \sqrt{\frac{E_k}{P_{rec} T_{PL}}} \mathbf{u}, \quad i = 2, \dots, n-1 \end{aligned} \quad (3.1)$$

where

$$\mathbf{u} = \frac{\mathbf{v}_n - \mathbf{v}_1}{\|\mathbf{v}_n - \mathbf{v}_1\|} \quad (3.2)$$

and T_{PL} can be found from

$$\mathbf{v}_n = \mathbf{v}_1 + \sum_{i=1}^{n-1} \sqrt{\frac{E_i}{P_{rec} T_{PL}}} \mathbf{u}, \quad (3.3)$$

i.e.,

$$T_{PL} = \frac{1}{P_{rec} \|\mathbf{v}_n - \mathbf{v}_1\|^2} \left(\sum_{i=1}^{n-1} \sqrt{E_i} \right)^2. \quad (3.4)$$

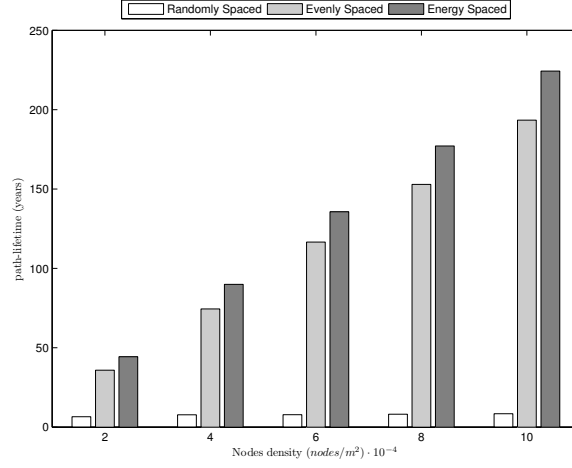
The obtained positions \mathbf{v}_i guarantee that the energy consumption is at a minimum for each node in the data flow. Thus, nodes are closer or further from the following neighbour depending on their residual energies. In [24] we set up a simulation scenario and compared the performance of random placement, evenly spaced placement and energy spaced placement in terms of path-lifetime, defined as the value of the minimum time duration that a node can be active in the current data flow, until its battery is completely out of energy, as in [94]. Evaluation parameters and results are reported also here in Table 3.1 and Fig. 3.1, respectively.

Bidirectional data flow

When the data flow is bidirectional, nodes spend energy to send data both in the backward and in the forward direction. Hence, their placement should vary

Table 3.1. Evaluation Parameters for Placement Algorithms without Nodes Mobility

Field Area ($L \times L$)	1000 m x 1000 m
Nodes Density (ρ)	$[2 \div 10] \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$
Source position (\mathbf{v}_1)	(0,0)
Destination position (\mathbf{v}_n)	(1000,1000)
Residual Energy Range (E_i)	15 ÷ 20 J
Maximum Transmission Radius (r_{max})	$1/(2\sqrt{\rho})$ m
Transmission Rate (r_T)	1 kb/s
Number of run for each scenario	100

**Fig. 3.1.** Path Lifetime for the three schemes.

accordingly. Unfortunately, the relaxation of monodirectionality assumption implies that the solution can not be derived in close-form. In [43] we stated the bidirectional optimal placement problem as follows:

$$\begin{aligned}
 & \max \min_i T_i \\
 & s.t. : \\
 & T_1 \leq \frac{E_1}{P_{rec} \|\mathbf{v}_1 - \mathbf{v}_2\|^2} \\
 & T_i \leq \frac{E_i}{P_{rec} (\|\mathbf{v}_i - \mathbf{v}_{i+1}\|^2 + \|\mathbf{v}_i - \mathbf{v}_{i-1}\|^2)}, \quad 1 < i < n \\
 & T_n \leq \frac{E_n}{P_{rec} \|\mathbf{v}_n - \mathbf{v}_{n-1}\|^2}
 \end{aligned} \tag{3.5}$$

where T_i is a function of the distances between node i and its neighbours. The proposed model is of max-min type with nonlinear constraints [95]. The objective consists of the maximization of the shortest node lifetime.

We called this model simply “bidirectional” since it is an extension of the “monodirectional” energy spaced model presented in [24]. It allows us to find the best placement when the energy of the nodes is used for both the forward and the backward flows.

A different approach is given by the “bidirectional energy-splitable” (BES) model also presented in [43]:

$$\begin{aligned}
 & \max \Gamma \\
 & s.t. \\
 & \Gamma \leq T_i, \quad 1 \leq i \leq n \\
 & T_1 \leq \frac{E_1}{P_{rec} \|\mathbf{v}_1 - \mathbf{v}_2\|^2} \\
 & T_i \leq \frac{\lambda * E_i}{P_{rec} \|\mathbf{v}_i - \mathbf{v}_{i+1}\|^2}, \quad 1 < i < n \\
 & T_i \leq \frac{(1-\lambda) * E_i}{P_{rec} \|\mathbf{v}_i - \mathbf{v}_{i-1}\|^2}, \quad 1 < i < n \\
 & T_n \leq \frac{E_n}{P_{rec} \|\mathbf{v}_n - \mathbf{v}_{n-1}\|^2}
 \end{aligned} \tag{3.6}$$

where there are two different constraints for the lifetime of the generic relay node i , one for the backward and one for the forward flow and the residual energy of the device is split between the two directions depending on the factor λ . When $0 \leq \lambda < 0.5$ more energy is allocated for the forward flow, while with $0.5 < \lambda \leq 1$ the backward flow has higher priority. For $\lambda = 0$ or $\lambda = 1$ we have completely asymmetric flows in the forward and backward direction, respectively, while for $\lambda = 0.5$ the flow is completely symmetric. In the former case, the optimal placement coincides with the solution computed with the monodirectional model (3.1). Model (3.6) is more flexible than (3.5) and more suitable to different flows’ requirements. In fact, λ could be considered as a function of some flow parameter, such as the load, the delay or the jitter and the scheme would find the best nodes placement depending on the chosen λ .

In Fig. 3.2 we investigated the best solution, in terms of path-lifetime, for the BES model when λ varies and $\rho = 5 \cdot 10^{-4}$ nodes/m². In Fig. 3.3 we compared the three placement schemes for a completely symmetric ($\lambda=0.5$) bidirectional data flow in the same conditions of Table 3.1 and when the energy spaced scheme has been optimized through the BES model. Still the energy spaced scheme outperforms the other two schemes, but in this case the evenly spaced scheme almost equals the performance of the energy spaced scheme. Let us recall here that the evenly spaced scheme assumes the same residual energy for all the nodes, therefore the resulting placement is implicitly computed for completely symmetric bidirectional flows.

The main disadvantage of the bidirectional models is that a central management station is required in order to collect the current information about residual energy, the positions of the involved nodes and find their most energy-efficient placement.

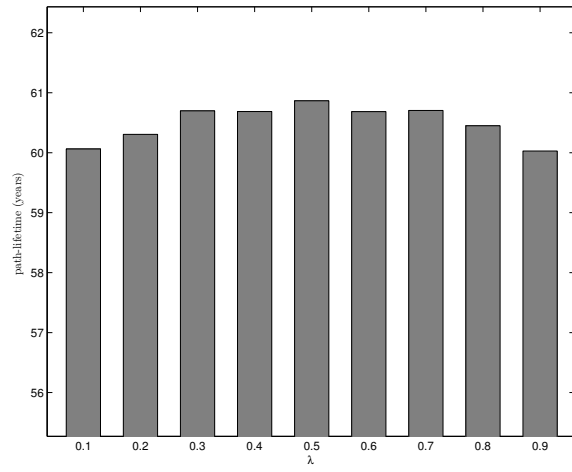


Fig. 3.2. Path Lifetime for the BES model when λ varies.

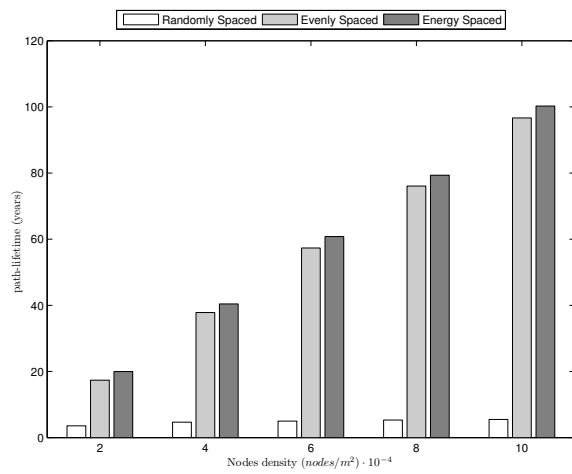


Fig. 3.3. Comparison of the lifetime for three placement schemes ($\lambda = 0.5$)

3.2.3 Real movement

In the previous section we showed that an energy-wise placement exists for both monodirectional and bidirectional data flows when different levels of nodes residual energy are considered. Since a human intervention is not possible every time that a new data flow occurs, the only viable approach for the nodes to reach their optimal positions is to let them self-organize in response to some event. Self-organization calls for a feasible technique that drives nodes to the optimal placement in a distributed and asynchronous way. A viable solution in this context is given by controlled mobility. In this section we discuss two controlled mobility algorithms for completely symmetric bidirectional data flows, based on the real movement of nodes towards the placements determined by the models of Section 3.2.2, in comparison with the random placement obtained in the deployment phase without any movement of the nodes. Since the routing problem is out of the scope of this work, we assume that, for both the algorithms, the routing protocol already chose the relay nodes which are initially located in the closest positions to the evenly spaced positions on the straight line between source and destination. The evenly spaced real movement algorithm (EvSRM), presented in [5], moves nodes towards the evenly spaced positions on the straight line between source and destination nodes. For the energy spaced real movement algorithm (EnSRM), we cannot use directly the placement computed by the mathematical framework in Section 3.2.2 because, as we said, that model requires a central management station. Thus, we conceived the algorithm in Fig. 1 that is distributed and uses only local information. At each iteration, each node involved in the data flow calculates, through EnSRM, the best backward and forward position to move using only the information on its position and its residual energy, and the positions and the residual energies of its closest neighbours. Then it physically moves to the central position between the computed backward and forward positions. Besides being distributed, EnSRM is also asynchronous, because each node runs the algorithm independently from any time reference. The Algorithm 1 considers EnSRM at the generic iteration k for the generic node i .

In order to determine the new position in an asynchronous and distributed way, EnSRM uses two coefficients: $\overrightarrow{\gamma}_i$ and $\overleftarrow{\gamma}_i$ that take into consideration the residual energy ratios of the node i in the forward and backward directions, respectively. Furthermore the movement of the node is damped, in order to avoid large oscillations, by the damping factor g as in the EvSRM algorithm [5].

In Fig. 3.4 we can see the tracking of nodes when they are running EvSRM (a) and EnSRM (b). In both the figures, the full black circles represent the nodes chosen by the routing algorithm, the full grey circles indicate the evenly (a) and energy (b) spaced positions, while the empty grey circles in (b) are the evenly spaced positions. From this figure, we have an example of a path travelled along by the nodes, in order to reach the evenly or energy spaced

Algorithm 1 Energy Spaced Real Movement Algorithm

$v_i(k)$: position of node i ;
 $v_{i-1}(k)$: position of node $i - 1$;
 $v_{i+1}(k)$: position of node $i + 1$;
 $E_i(k)$: residual energy of node i ;
 $E_{i-1}(k)$: residual energy of node $i - 1$;
 $E_{i+1}(k)$: residual energy of node $i + 1$;
 $\vec{v}_i(k)$: temporary variable for storing the new position of node i in the forward direction;
 $\overleftarrow{v}_i(k)$: temporary variable for storing the new position of node i in the backward direction;
 $\vec{\gamma}_i(k)$: residual energy coefficient of node i in the forward direction;
 $\overleftarrow{\gamma}_i(k)$: residual energy coefficient of node i in the backward direction;
repeat
 send $v_i(k)$ to neighbours $i - 1$ and $i + 1$;
 send $E_i(k)$ to neighbours $i + 1$
 receive $v_{i-1}(k)$, $v_{i+1}(k)$, $E_{i-1}(k)$ and $E_{i+1}(k)$;
 set $\vec{\gamma}_i(k) = \frac{\sqrt{E_{i-1}(k)}}{\sqrt{E_{i-1}(k)} + \sqrt{E_i(k)}}$;
 set $\overleftarrow{\gamma}_i(k) = \frac{\sqrt{E_{i+1}(k)}}{\sqrt{E_{i+1}(k)} + \sqrt{E_i(k)}}$;
 set $\vec{v}_i(k) = ((1 - \vec{\gamma}_i(k))v_{i-1}(k) + \vec{\gamma}_i(k)v_{i+1}(k))$;
 set $\overleftarrow{v}_i(k) = (\overleftarrow{\gamma}_i(k)v_{i-1}(k) + (1 - \overleftarrow{\gamma}_i(k))v_{i+1}(k))$;
 set $v'_i(k) = (\vec{v}_i(k) + \overleftarrow{v}_i(k))/2$;
 move to $v''_i(k) = v_i(k) + g(v'_i(k) - v_i(k))$;
 $v_i(k + 1) = v''_i(k)$
until (convergence condition is satisfied)

positions on the straight line between the terminal nodes of a bidirectional data flow. In Section 3.2.5 we will show the energy spent for travelling along such a winding path. In the following subsections, first we will analyze the EnSRM scheme by giving the proof of convergence and connectivity; secondly we will present EnSRM and EvSRM behavior in terms of nodes residual energy, travelled distances and number of iterations, when the damping factor varies; finally we will compare the two schemes which use real movement of nodes with the random placement scheme, in order to substantiate the need for the virtualization of nodes' movements.

Convergence and connectivity**Convergence**

Assume that the number of nodes involved in the data flow between the source node S and the destination node D is $n-2$ and that the position of node i at the k^{th} iteration of the algorithm is $\mathbf{v}_i(k)$. Let γ_i be the ratio

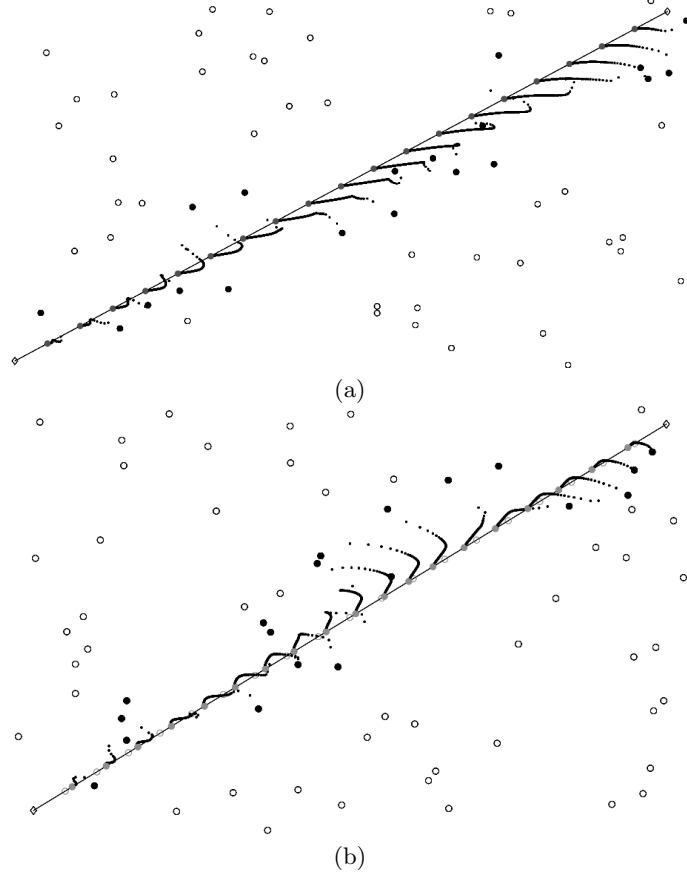


Fig. 3.4. a) Tracking of nodes movement for the EvSRM scheme and b) the EnSRM scheme

$\gamma_i(k) = \frac{\sqrt{E_{i-1}(k)}}{\sqrt{E_{i-1}(k)} + \sqrt{E_i(k)}}$ at the k^{th} iteration. The update equation for node i is:

$$\begin{aligned} \mathbf{v}_i(k+1) = & \mathbf{v}_i(k) + g[(1 - \gamma_i(k))\mathbf{v}_{i-1}(k) + \\ & + \gamma_i(k)\mathbf{v}_{i+1}(k) - \mathbf{v}_i(k)], k = 1, 2, \dots \end{aligned} \quad (3.7)$$

where $g \in (0, 1]$ is called damping factor and it is used as in [5] for avoiding possible oscillations.

Let the position error be:

$$\mathbf{e}_i(k) = \mathbf{v}_i(k) - \mathbf{v}_i, \quad i = 1, 2, \dots, n. \quad (3.8)$$

Where \mathbf{v}_i is the optimal position of node i calculated in (3.1). In order to show the convergence of our algorithm we will show that the position error will go to zero by iterating (3.7). At this point we observe that:

$$\mathbf{v}_{i-1}(k) = \mathbf{e}_{i-1}(k) + \mathbf{v}_i - \sqrt{\frac{E_{i-1}(k)}{P_{rec}T_{PL}}}\mathbf{u}. \quad (3.9)$$

and

$$\mathbf{v}_{i+1}(k) = \mathbf{e}_{i+1}(k) + \mathbf{v}_i + \sqrt{\frac{E_i(k)}{P_{rec}T_{PL}}}\mathbf{u}. \quad (3.10)$$

The error at the $(k+1)^{th}$ iteration is:

$$\begin{aligned} \mathbf{e}_i(k+1) &= \mathbf{v}_i(k+1) - \mathbf{v}_i \\ &= \mathbf{v}_i(k) + g[(1 - \gamma_i(k))\mathbf{v}_{i-1}(k)] + \\ &\quad + g[\gamma_i(k)\mathbf{v}_{i+1}(k) - \mathbf{v}_i(k)] - \mathbf{v}_i \\ &= (1 - g)\mathbf{e}_i(k) + g[(1 - \gamma_i(k))\mathbf{e}_{i-1}(k) + \\ &\quad - (1 - \gamma_i(k))\sqrt{\frac{E_{i-1}(k)}{P_{rec}T_{PL}}}\mathbf{u} \\ &\quad + \gamma_i(k)\sqrt{\frac{E_i(k)}{P_{rec}T_{PL}}}\mathbf{u} + \gamma_i(k)\mathbf{e}_{i+1}(k)] \end{aligned} \quad (3.11)$$

In order to establish if the error goes to zero in a limited number of iterations, we introduce a vector $\mathbf{E}(k) = (\mathbf{e}_1(k), \mathbf{e}_2(k), \dots, \mathbf{e}_n(k))$. In this way we obtain:

$$\begin{aligned} \mathbf{e}_i(k+1) &= g(1 - \gamma_i(k))\mathbf{e}_{i-1}(k) + \\ &\quad + (1 - g)\mathbf{e}_i(k) + g\gamma_i(k)\mathbf{e}_{i+1}(k). \end{aligned} \quad (3.12)$$

Using a matrix notation, it is possible to rewrite (3.12) as $\mathbf{E}(k+1) = \mathbf{A}(k+1)\mathbf{E}(k) = (\prod_{i=1}^{k+1} \mathbf{A}(i))\mathbf{E}(0)$ for each iteration, where $\mathbf{A}(k)$ is the matrix:

$$\begin{pmatrix} 1 - g & g\gamma_1(k) & 0 & 0 & 0 \\ g(1 - \gamma_2(k)) & (1 - g) & g\gamma_2(k) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & g(1 - \gamma_i(k))(1 - g) & g\gamma_i(k) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & g(1 - \gamma_n(k))(1 - g) & 0 \end{pmatrix} \quad (3.13)$$

Theorem 1 *Matrix \mathbf{A} constant. If the entries of the matrix \mathbf{A} are constant (i.e. the γ_i coefficients are constant) then the mobility scheme will drive nodes to the stable configuration in (3.1).*

Proof: The matrix we obtained is classified as a tri-diagonal matrix. Since the entries of our matrix \mathbf{A} are all non-negatives, we can exploit the Perron-Frobenius theorem [96]. Let R_i and T_j denote the sum of the absolute values of the entries in the i^{th} row and the sum of the absolute values of the entries in the j^{th} column, respectively, so that $R = \max_i R_i$, $T = \max_j T_j$, then Perron-Frobenius theorem states the first inequality in the following:

$$\lambda_i \leq \min(R, T) \leq 1, \quad (3.14)$$

where the second inequality comes from the maximum value of R_i for matrix \mathbf{A} .

Given (3.14) and applying a standard result in the matrix theory [97] we have $\lim_{k \rightarrow \infty} \mathbf{A}^k = 0$ which implies that $\lim_{k \rightarrow \infty} \mathbf{E}(k) = 0$. In practice, this means that the algorithm converges to the configuration given in (3.1), where nodes are positioned along the line between the source node S and the destination node D and their positions depend on their initial energies.

However, matrix \mathbf{A} is time-varying. Indeed, the γ_i coefficients could vary at each iteration of the algorithm. Hence, not so much can be said about the stability of the system. We studied the stability radius of the matrix \mathbf{A} by using the theory of matrix perturbations [93]. The stability radius expresses the “distance” of the system from the instability region. By using the values for $\gamma_i(k)$ taken from our simulations, we found out what it follows:

Theorem 2 *Matrix \mathbf{A} time-varying. The stability radius of Matrix \mathbf{A} is always greater than the largest perturbation that could affect the system.*

From (3.13) we can write \mathbf{A} at the iteration $k + 1$:

$$\begin{pmatrix} 1-g & g\gamma_1(k+1) & 0 & 0 & 0 \\ g(1-\gamma_2(k+1)) & (1-g) & g\gamma_2(k+1) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & g(1-\gamma_i(k+1)) & (1-g) & g\gamma_i(k+1) & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & g(1-\gamma_n(k+1)) & (1-g) \end{pmatrix} \quad (3.15)$$

Both matrices (3.13) and (3.15) are tri-diagonal and their elements are non-negative and smaller than 1. Furthermore, matrix \mathbf{A} is stable because of the Perron-Frobenius theorem and the result in [97] Therefore, we can consider $\mathbf{A}(k+1)$ as a perturbation of $\mathbf{A}(k)$:

$$\mathbf{A}(k+1) = \mathbf{A}(k) + \mathbf{B}\Delta\mathbf{C}, \quad (3.16)$$

where \mathbf{B} and \mathbf{C} are non-negative matrices and Δ is the perturbation matrix:

$$\Delta = \left(\begin{array}{c} [p_{11}\delta_{11} \ \dots \ p_{1n}\delta_{1n}] \\ \vdots \\ [p_{n1}\delta_{n1} \ \dots \ p_{nn}\delta_{nn}] \end{array} : p_{ij} \geq 0 \right). \quad (3.17)$$

Hence, we apply the results in [93], which allow us to calculate the stability radius of the system:

$$r_R^d = \frac{1}{\rho(\mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{P})}, \quad (3.18)$$

where $\rho(\cdot)$ denotes the spectral radius. In our case \mathbf{B} and \mathbf{C} are the identity matrix, \mathbf{P} gives the tri-diagonal structure to the perturbation:

$$\mathbf{P} = \begin{pmatrix} p_{11} & \dots & p_{1n} \\ \vdots & \vdots & \vdots \\ p_{n1} & \dots & p_{nn} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad (3.19)$$

thus resulting in the perturbation matrix (for the sake of simplicity we write Δ as a 3-by-3 matrix):

$$\Delta = \left(\begin{array}{c} \begin{bmatrix} 0 & \delta_1 & 0 \\ -\delta_2 & 0 & \delta_2 \\ 0 & -\delta_3 & 0 \end{bmatrix} \\ : \delta_i = g(\gamma_i(k+1) - \gamma_i(k)) \end{array} \right) \quad (3.20)$$

Authors in [93] also define:

$$\|\Delta\| := \max_{i,j} \{|\delta_{ij}| : p_{ij} \neq 0\} \quad (3.21)$$

in order to determine the entry of Δ which causes the largest perturbation. If the stability radius is not greater than the largest perturbation, then the perturbation would lead the system to an instability region.

We analyzed the stability radius for our simulated scenarios by introducing the perturbations of the γ_i coefficients. Let us recall the definition of $\gamma_i(k) = \frac{\sqrt{E_{i-1}(k)}}{\sqrt{E_{i-1}(k)} + \sqrt{E_i(k)}}$. We define the difference between the energy coefficient of node i at the iteration $k+1$ and k as $\Delta\gamma_i \triangleq \gamma_i(k+1) - \gamma_i(k)$. In our simulations, the initial energy of nodes is in the range $15 \div 20$ J, therefore for the energy coefficients at the first iteration, it stands $0.4641 \leq \gamma_i(1) \leq 0.5359 \forall i$. We extensively studied the evolution of these coefficients, when the damping factor varies, in order to determine the distance of the system from the instability. In Fig. 3.5a and 3.5b we reported the plot of two groups of γ_i that show a similar behaviour.

As we can see, the value of γ_i increases or decreases quickly and after few iterations it converges to a constant value. These plots have been drawn for $g = 0.75$, but the same behaviour has been verified for smaller (and larger) values of the damping factor on a larger (and smaller) number of iterations.

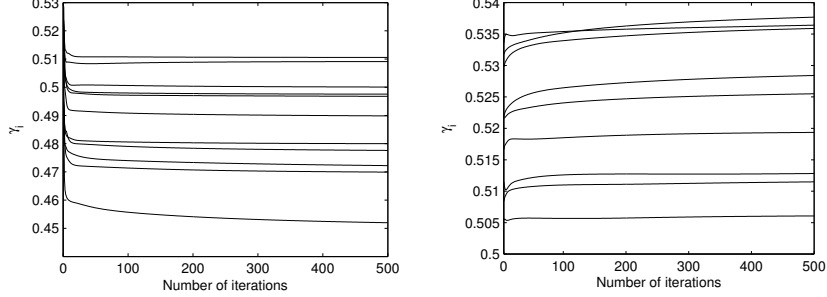


Fig. 3.5. Evolution of two groups of γ_i with the number of performed iterations for $g = 0.75$.

In Table 3.2 we reported the values of the number of iterations needed in a scenario with high nodes density ($10 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$) to satisfy the convergence condition ($10^{-2}m$ of movement in the last iteration) when g varies. Furthermore, we reported the maximum value of $\Delta\gamma_i$ registered on all the iterations. In the following column, we reported the same value multiplied by the damping factor, because it represents the largest perturbation that will be compared to the stability radius, which is in the last column. As we can see the stability radius calculated as in (3.18) from the simulation data is greater than the largest perturbation for all the values of the damping factor. The number of iterations for achieving a stable value (variation from the value at the previous iteration smaller than $1 \cdot 10^{-4}$) varies depending on the damping factor as in Table 3.2. Number of iterations and $\max \Delta\gamma_i$ have been statistically averaged over 100 runs.

Table 3.2. Energy coefficients at the convergence

g	Number of iterations	$\max \Delta\gamma_i$	$g \cdot \max \Delta\gamma_i$	r_R^d
0.25	212	0.0093	0.002325	0.019
0.5	159	0.0167	0.00835	0.039
0.75	144	0.0286	0.02145	0.057
1	126	0.0303	0.0303	0.075

Connectivity

Theorem 3 *By using the mobility scheme presented in Algorithm 1, the connectivity is not lost at each iteration for all the meaningful cases.*

Proof: Without loss of generality we assume that all the nodes involved in the data flow at the generic iteration k are connected.

$$\begin{cases} \|\mathbf{v}_i(k) - \mathbf{v}_{i-1}(k)\|^2 \leq r_{max}^2 \\ \|\mathbf{v}_i(k) - \mathbf{v}_{i+1}(k)\|^2 \leq r_{max}^2 \end{cases} \quad (3.22)$$

Where r_{max} , defined in Table 3.1, is the maximum transmission radius allowed in order for a data packet to be considered correctly delivered to the destination node.

When the iteration $(k+1)$ for node \mathbf{v}_i is considered, the connectivity condition becomes:

$$\begin{cases} \|\mathbf{v}_i(k+1) - \mathbf{v}_{i-1}(k)\|^2 \leq r_{max}^2 \\ \|\mathbf{v}_i(k+1) - \mathbf{v}_{i+1}(k)\|^2 \leq r_{max}^2 \end{cases} \quad (3.23)$$

The proof that connectivity is preserved at each iteration is based on the update equation:

$$\begin{aligned} \mathbf{v}_i(k+1) = & \mathbf{v}_i(k) + g[(1 - \gamma_i(k))\mathbf{v}_{i-1}(k) + \\ & + \gamma_i(k)\mathbf{v}_{i+1}(k) - \mathbf{v}_i(k)]. \end{aligned} \quad (3.24)$$

So that (3.23) becomes:

$$\begin{cases} \|\mathbf{v}_i(k) + g[(1 - \gamma_i(k))\mathbf{v}_{i-1}(k) + \\ + \gamma_i(k)\mathbf{v}_{i+1}(k) - \mathbf{v}_i(k)] - \mathbf{v}_{i-1}(k)\|^2 \leq r_{max}^2 \\ \|\mathbf{v}_i(k) + g[(1 - \gamma_i(k))\mathbf{v}_{i-1}(k) + \\ + \gamma_i(k)\mathbf{v}_{i+1}(k) - \mathbf{v}_i(k)] - \mathbf{v}_{i+1}(k)\|^2 \leq r_{max}^2 \end{cases} \quad (3.25)$$

Working on the left-hand sides we have:

$$\begin{cases} \|(1 - g + g\gamma_i(k))\mathbf{x}_i(k) + g\gamma_i(k)\mathbf{y}_i(k)\|^2 \leq r_{max}^2 \\ \|g(\gamma_i(k) - 1)\mathbf{x}_i(k) + (g\gamma_i(k) - 1)\mathbf{y}_i(k)\|^2 \leq r_{max}^2 \end{cases} \quad (3.26)$$

where $\mathbf{x}_i(k) = \mathbf{v}_i(k) - \mathbf{v}_{i-1}(k)$ and $\mathbf{y}_i(k) = \mathbf{v}_{i+1}(k) - \mathbf{v}_i(k)$ (see Fig. 3.6). Since both $\|\mathbf{x}_i(k)\| \leq r_{max}$ and $\|\mathbf{y}_i(k)\| \leq r_{max}$, we can re-write (3.26) as:

$$\begin{cases} (r_{max}^2[(1 - g + g\gamma_i(k))^2 + g^2\gamma_i(k)^2 + \\ + 2(1 - g + g\gamma_i(k))g\gamma_i(k)\cos\theta_i(k)] \leq r_{max}^2) \\ (r_{max}^2[(g(\gamma_i(k) - 1))^2 + (g\gamma_i(k) - 1)^2 + \\ + 2g(\gamma_i(k) - 1)(g\gamma_i(k) - 1)\cos\theta_i(k)] \leq r_{max}^2) \end{cases} \quad (3.27)$$

where $\theta_i(k)$ is the angle between $\mathbf{x}_i(k)$ and $\mathbf{y}_i(k)$ as in Fig. 3.6. By solving (3.27) we found that g can range in the intervals:

$$\begin{cases} \left(0, \frac{2(1-\gamma_i(k)-\gamma_i(k)\cos\theta_i(k))}{1-2\gamma_i(k)+2\gamma_i(k)^2-2\gamma_i(k)\cos\theta_i(k)+2\gamma_i(k)^2\cos\theta_i(k)}\right), \\ \left(0, \frac{2(\gamma_i(k)+\gamma_i(k)\cos\theta_i(k)-\cos\theta_i(k))}{1-2\gamma_i(k)+2\gamma_i(k)^2-2\gamma_i(k)\cos\theta_i(k)+2\gamma_i(k)^2\cos\theta_i(k)}\right) \end{cases} \quad (3.28)$$

By imposing that the upper bound of the intervals (3.28) must be positive, which guarantees that $g > 0$, we can find the sufficient conditions on $\gamma_i(k)$ and $\theta_i(k)$ to preserve connectivity:

$$\left\{ \begin{array}{l} 0 \leq \gamma_i(k) \leq 0.5, 0 \leq \cos \theta_i(k) \leq \left(\frac{\gamma_i(k)}{1-\gamma_i(k)} \right) \\ \cup \\ 0.5 \leq \gamma_i(k) \leq 1, 0 \leq \cos \theta_i(k) \leq \left(\frac{1-\gamma_i(k)}{\gamma_i(k)} \right) \end{array} \right. \quad (3.29)$$

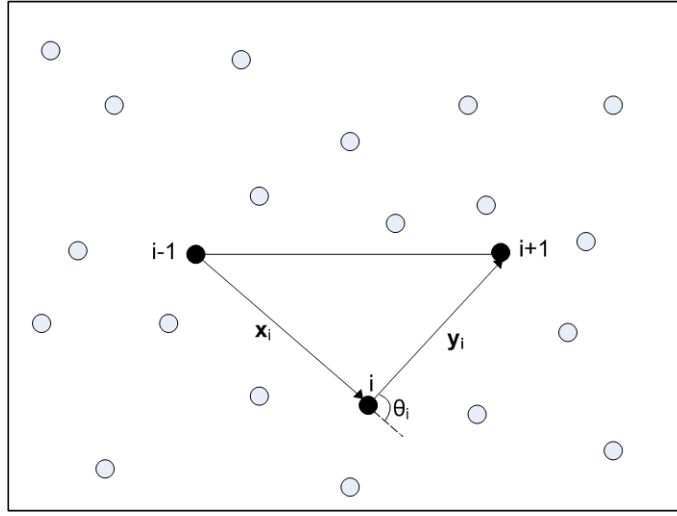


Fig. 3.6. Conditions for preserving the connectivity.

The shadowed area in Fig. 3.7 represents the region of values for $\gamma_i(k)$ and $\cos \theta_i(k)$ that guarantees that the connectivity is not lost.

As we can see in all the meaningful cases (i.e.: $0 \leq \gamma_i(k) \leq 1, -1 \leq \cos \theta_i(k) \leq 0$) the connectivity is guaranteed. The region $0 \leq \cos \theta_i(k) \leq 1$ includes unwanted values, because it represents a case where the connecting nodes depart significantly from the main direction of the source-destination link. Since we had to use an upper bound for $\|\mathbf{x}_i(k)\|$ and $\|\mathbf{y}_i(k)\|$, we cannot conclude anything on the region of values of $\gamma_i(k)$ and $\cos \theta_i(k)$ that are outside the shadowed region but in our extensive simulation experiments the connectivity has never been lost.

Dependence from the damping factor

Both the Algorithm 1 in this section and the algorithm in [5] use the damping factor g , when the nodes have to move to a new position. This factor varies in

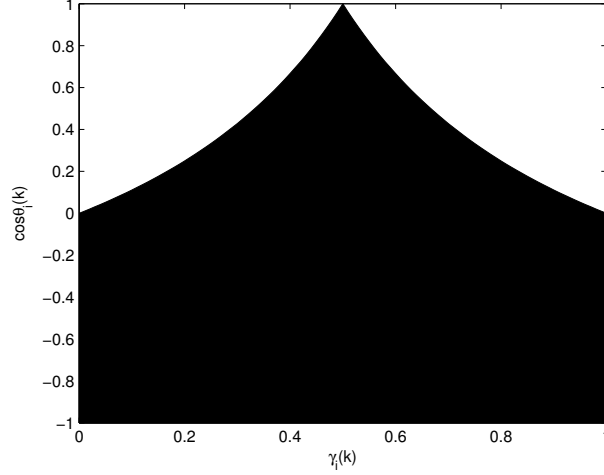


Fig. 3.7. Admissible region of values for $\gamma_i(k)$ and $\cos\theta_i(k)$.

the interval $(0,1]$ and is used to avoid that nodes oscillates around the straight line between source and destination. In this subsection we want to illustrate the effects of the damping factor on the performance of the algorithms. In particular, by using the same simulation scenario of Section 3.2.2 and the parameters listed in Table 3.3 (with $\rho = 6 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$ and $l = 600\sqrt{2}m$), we simulated the behavior of EvSRM and EnSRM when the damping factor varies, in order to understand the impact of this factor on the performance of the two algorithms. Specifically, we investigate the role of the damping factor on residual energy, travelled distance and number of iterations to reach the optimal placement.

Table 3.3. Evaluation Parameters for Placement Algorithms by Using Nodes Mobility

Field Area ($L \times L$)	$1000m \times 1000m$
Nodes Density (ρ)	$[2 \div 10] \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$
Flow Time Length (T_F)	$87.6 \cdot 10^3 h$
Source-Destination Distance (l)	$[200 \div 1000] \sqrt{2}m$
Maximum Transmission Radius (r)	$1/(2\sqrt{\rho}) m$
Relay Nodes Number (N)	l/r
Initial Residual Energy Range (E_i)	$15 \div 20 J$
Minimum Required Power (P_{rec})	$1.0 \cdot 10^{-11} W/m^2$
Transmission Rate (r_T)	$1 kb/s$
Movement Constant (m)	$0.1 J/m$
Damping Factor (g)	$[0.25 \div 1]$
Number of runs for each scenario	100
Statistical confidence interval	95%

In Fig. 3.8a we can see the average residual energy of the nodes after a data flow of duration T_F has been forwarded from the source to the destination node. The energy expenditure of the nodes is composed of two contributions:

- Relaying data: as already mentioned in Section 3.2.2, the energy model used for data transmission is taken from [91].
- Physical movement: the energy consumed for travelling a distance d is assumed to be proportional to the travelled distance by a constant factor m that takes into account only the dynamic friction, as in [163]. A more accurate energy model would take into consideration also the energy spent by nodes to win the static friction, every time that they stop and move.

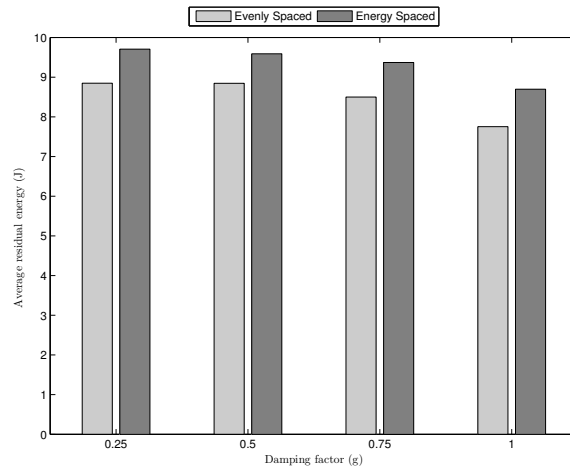
The impact of damping nodes' movement on the energy performance is remarkable. When nodes' movement is highly damped ($g = 0.25$), the energy saved is up to 11% for EnSRM and 14% for EvSRM in respect of no damping ($g = 1$). The difference of the energy saved for the two schemes is due to a general quicker convergence of the EnSRM algorithm in respect of the EvSRM. The same effect is evident also in the average travelled distance diagram (Fig. 3.8b). When the movement is not damped, nodes move 10 m for EvSRM and 8 m for EnSRM more on average than when movement is highly damped, which means 16% and 13% of more movements, respectively. Finally, damping the movement has the largest impact on the number of iterations needed for the nodes to converge to the optimal placement in Fig. 3.8c. In these simulations, the convergence condition is satisfied, for each node, when it moved less than $1 \cdot 10^{-6}m$ in the last iteration. When nodes are free to move to the computed positions, it takes around 80% (83% for EvSRM and 81% for EnSRM) less time to reach the optimal positions than when nodes' movement is highly damped. From these results, we understand that the damping factor could be tuned in order to meet the time constraints required for the data flow. When the data flow does not generate large quantities of data or does not have strict time constraints, the damping factor can be low in order to save energy and move nodes slowly. On the contrary, for tight schedules or large quantity of data to relay, it is convenient to move nodes to their optimal positions as quick as possible in order to meet the constraints and save energy for the communication.

3.2.4 Virtual movement

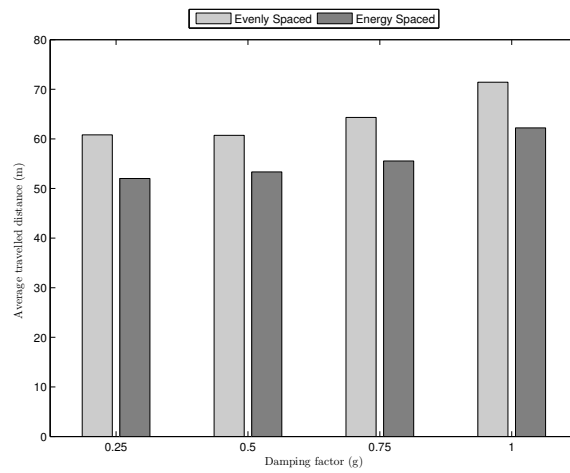
Motivations

In this section, we discuss the reasons why we consider nodes mobility as a profitable feature in WSN, but, at the same time, we want to illustrate what is required by a movement scheme in order to guarantee that the mobility does not become wasteful in terms of network's resources.

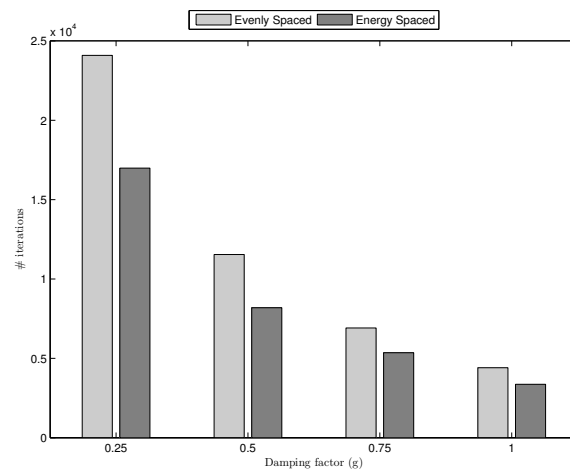
The first motivation is general and concerns the advantages of controlled mobility in WSN, while the second and the third are more specific and regard



(a)



(b)



(c)

Fig. 3.8. a) Nodes residual energies b) Average travelled distances c) Number of iterations needed to achieve the optimal placement when g varies

the optimality of some nodes placement and the correct usage of controlled mobility for achieving those placements, respectively. Thus, in the second motivation we show the performances of two placement schemes, without taking into consideration the movement of the nodes:

1. evenly spaced scheme, which places the nodes along the straight line between the terminal nodes, in evenly spaced positions [5];
2. energy spaced scheme, which places the nodes along the straight line between the terminal nodes, according to the different levels of nodes' residual energies [24].

In [5], the authors also introduce a movement algorithm for the nodes. It has been comparatively evaluated with the version modified by taking into account the residual energy of the nodes, according to the energy spaced scheme, in order to illustrate the third motivation.

To substantiate and illustrate our arguments, we also set up an evaluation environment. We perform the simulations over a 1000m x 1000m deployment field consisting of a variable number of nodes, distributed randomly over the field. A bidirectional data flow is activated between a pair of static terminal nodes and a geographical routing algorithm chooses the nodes that are closest to the evenly spaced positions, over the straight line between the terminal nodes. The effect of placement and mobility on the network is also studied by investigating scenarios with nodes density of 2, 4, 6, 8 and $10 \cdot 10^{-4} \text{nodes}/\text{m}^2$, respectively. For each simulation, we run 100 experiments with different seeds and take the average of the measured values, in order to reach a confidence interval of 95%. The details of the energy model and the simulation environment can be found in Section 3.2.5.

Advantages of controlled mobility

The first motivation for using controlled mobility is that it enriches the communication paradigm with a new design dimension. Controlled mobility moves the focus from the best node for performing a task to the best node in the optimal position. As we already saw, mobility of devices can improve the most important performance parameters: coverage, capacity and energy consumption. More specifically sensor networks are deployed in order to monitor a certain geographical area or some interesting places. Hence, nodes must have the capacity to communicate with each other, through a multi-hop transmission, thus creating a fully connected network. Two main disadvantages of a network constituted only by static nodes are: the unadaptability to the dynamics of the environment and to the local disconnections, the vulnerability of the fixed positions of the nodes to a malicious attacker. Intuitively, by modifying the positions of the nodes it is possible to optimize and extend the coverage and connectivity of the network and overcome the cited disadvantages. Also regarding the network capacity, the controlled mobility can highly improve the performance. Both the “data mules”, whose mobility is also used

for improving the delivery ratio of data and, in delay-tolerant networks, the “message ferrying” exploit controlled mobility in order to achieve the same task of transporting data with a high delivery ratio, also where end-to-end paths do not exist between nodes. Concerning energy consumption and efficiency, which are the focus of this work, we discuss them in the two following subsections.

Existence of optimal placements

Most of the schemes that improve coverage and capacity of a sensor network depend on the continuous movement of nodes in the field. Regarding the energy consumption, some analytical schemes have been proposed in order for the nodes to move only between their original position and the optimal one. Through these schemes, it is possible to improve the energy efficiency of the network by using controlled mobility. In this subsection, we want only to show the advantages of the final optimal placement, in respect of a random deployment of nodes in the field, without taking into consideration the energy spent by the movement of nodes. In Fig. 3.9, the two cited placement schemes are compared with the random one, when nodes density increases. In order to calculate the lifetime of the path of nodes involved in a bidirectional data flow, we assume that the data flow has an infinite duration and when nodes start the relay, they have different residual energies. The results show that a better placement means a longer lifetime.

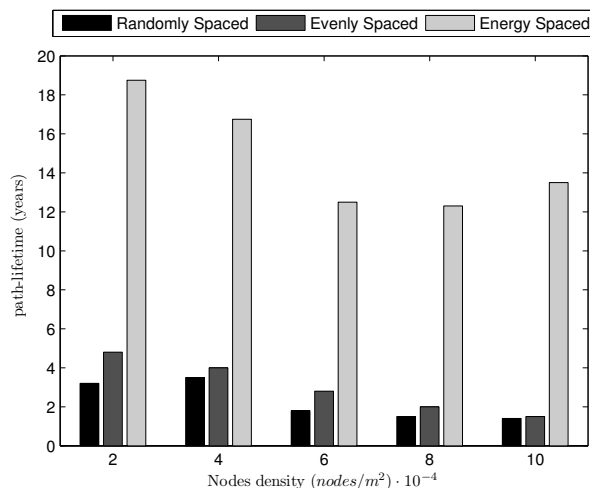


Fig. 3.9. Comparison of the lifetime for three placement schemes

In fact, both of the schemes improve the lifetime of the network, and the energy spaced one, by considering different residual energy levels, outperforms

the other. Thus, the second motivation for designing a controlled mobility scheme is that it is possible to compute optimal placement of the nodes for some specific target, and controlled mobility is the way to achieve it. On the other hand, a specific placement could be attained directly in the nodes deployment phase, but it would hardly be practical, expensive and not reactive to the environment changes. Since our focus is on the energy efficiency, we used two schemes that aim to lifetime maximization, but mathematical formulations could be contrived also for different targets.

Energy consumption with mobile sensors

In this last subsection, we want to show the effects of moving the nodes from one position to another one. Continuous or uncontrolled movements of the sensor nodes introduce the side effect of wasting more energy than the nodes would save by reaching different positions. Our concern is to show that, also when the movement is controlled, the trajectories followed by the nodes for arriving at the best positions can be energy inefficient. In Fig. 3.10 (a), the full blue circles represent the nodes chosen by the routing algorithm, the full green circles indicate the energy spaced positions, while the empty green circles are the evenly spaced ones. In the same figure, we have an example of a winding path travelled along by the nodes, in order to reach the energy spaced positions, on the straight line between the terminal nodes of a bidirectional data flow. As shown in 3.2.3, the movement algorithm makes each node, in each iteration, move to the position computed according to the residual energy of the previous and the following nodes in the path. The same figure could be plotted also for the evenly spaced algorithm, which moves each node, iteration by iteration, to the central position between its two most adjacent nodes in the path. In Fig. 3.10 (b) the residual energies of the nodes, after they completed the movement and the transmission phase, are shown. We recall that both the cited mechanisms are completely distributed, use only local information and allow nodes to work together toward the common purpose of achieving the wanted placement. A centralized scheme that could use the complete knowledge of nodes positions and energies would determine the optimal placement much quicker, but it would show all the disadvantages of this kind of approach. In both the mechanisms the movement is damped by the damping factor g , which is a multiplicative constant that varies in the interval $[0, 1]$. The initial values of energy, damping factor and transmission flow time-length are listed in Table 3.3. We compare the two schemes of placement and movement with the case that nodes did not move at all from their initial random positions, in terms of residual energy. It is possible to see that the movement makes the nodes have a lower level of residual energy. In respect of the comparison of the previous subsection, here the duration of the flow is finite. This more realistic assumption allows us to study the network when dynamic changes in the environment happen. Apparently, this suggests us not to move the nodes, in order to react better to birth and death of the flows or other changes in the

network. This conclusion will be contrasted by the introduction of our new scheme in the next section.

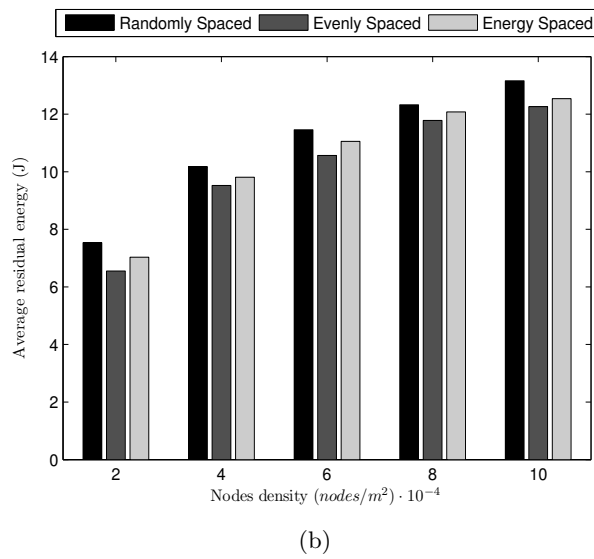
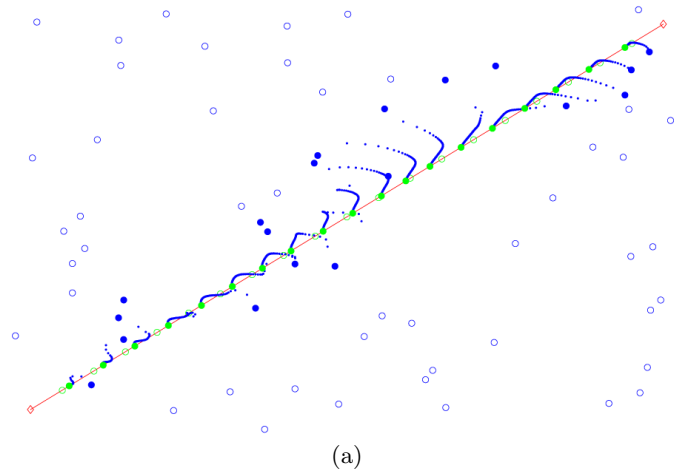


Fig. 3.10. a) Tracking of nodes movement for the energy spaced scheme b) Nodes residual energies after the movement

Since the damping factor contributes to control the movement of nodes, we also performed simulation in order to discover how the residual energy and the number of iterations to reach a final placement vary, when g varies.

As we can see in Fig. 3.8, the residual energy (a) is higher when the movement is more damped, because oscillations are attenuated and the average travelled distances are shorter (b), but this leads to a slower convergence (c). The purpose of these plots is to show that a new scheme that adds up both a quick convergence and a good energy efficiency has to be designed.

In conclusion, the expected features that a mobility mechanism should possess are: distributed, quickly convergent, adaptative, energy efficient and suitable to be used in a self-organizing network. From the explained motivations, we are pushed to think that the only viable solution is represented by controlled mobility, which intrinsically requires the cooperation among the nodes, thus creating the right set for developing a distributed scheme in the self-organizing systems context. In the following section, we will introduce a new mechanism based on controlled mobility that will be shown to exhibit all the cited features.

Virtual Movement Scheme

In this section we introduce our scheme of placement and movement of nodes. This mechanism will be tested and evaluated in order to be used for sensors applications with energy constraints, although it could be used for other objectives in different distributed, self-organizing contexts as well. The basic idea of the Virtual Movement Scheme is to use a distributed approach for calculating the nodes final positions which adds up the energy efficiency in the movement phase and the quickness of the centralized approach. By using the usual distributed approach, nodes reach their final positions after several iterations, which include movements and exchange of messages among the nodes. With the virtualization of the movements, after a sequence of iterations, which do not include any movement, the nodes will know their final positions, as if they were calculated by a central computational unit, and only then, will they move toward those positions.

The algorithm in [5], one of the first in WSN related with controlled mobility directed to energy saving, presents at least two improvements:

1. the optimal positions definition and determination;
2. the nodes movement toward the final positions.

Regarding the first, it has been shown in [24] that the more realistic assumption of nodes with different residual energies leads to a different placement, which remarkably improves the network lifetime. The virtualization of the movements focuses on the second improvement. As we have shown in the previous section, the movement of nodes towards the final positions follows trajectories that are expensive in terms of energy, in fact, the travelled distances are never the shortest between initial and final positions. In [5], one of the reasons for the introduction of the damping factor, g , is to attenuate the oscillations in the movement of the nodes. Although the abrupt oscillations are effectively smoothed, we have shown in the previous section that the

residual energies after the movement or the number of iterations to reach a stable placement still display an unsatisfying performance. The proposed algorithm improves the performance by making each node, ordered by a chosen direction from one static terminal node to the other, follow asynchronously this scheme:

1. The node computes its new position, which can be both evenly spaced or energy spaced. The position is calculated only by using local information: neighbours' positions and, for the energy spaced scheme, residual energies. This position is saved by the node and considered as a virtual position.
2. The node communicates its new virtual position to its closest neighbours in the path. In turn, the neighbour that follows in the chosen direction, will calculate its new virtual position depending on this new information. No movement of nodes has been performed in the network till this moment.
3. The algorithm for each node converges when the node calculates, as a new virtual position, the same position of the previous iteration. When this happens, the last saved virtual position is the optimal place, where the node can move following the best path from its initial and actual position.
4. We assume that all nodes move at the same velocity, so that, once the node that is the furthest from its optimal position completes its movement, it will trigger the beginning of the transmission phase. The furthest node can be determined by a simple election mechanism, usual in the distributed systems.

It is important to notice that the connectivity has to be guaranteed only for both the initial and the final positions, because intermediate placements are not allowed anymore. Furthermore, once the optimal position has been computed, it is possible to move the node following the trajectory that best suites the application's target. In our case, since we are interested in minimizing the energy consumption, and we will assume a planar, free of obstacle sensor field, the best path is the shortest path, which minimizes the travelled distances by making the node move along the straight line between the actual position and the virtual position.

The implementation of the algorithm for the Energy Spaced Virtual Movement scheme in pseudo-code follows:

As for the algorithm in section 3.2.3, this scheme computes both a forward and a backward position for each node, and the new virtual position is calculated as the mean between the two computed positions. Similarly, it is possible to provide the pseudo-code for the Evenly Spaced Virtual Movement scheme.

In Fig. 3.11 we can see a simulation of the algorithm Energy Spaced Virtual Movement. This plot uses the same symbology of Fig. 3.10 (a), with which it must be compared in order to see the different trajectories. With the Energy Spaced Virtual Movement, the movements are not performed at each iteration, but once for all as a last step of the algorithm. Thus, in the Fig. 3.11 it is shown in blue the straight and shortest path performed by the nodes.

Algorithm 2 Energy Spaced Virtual Movement

x_i : current position of node i ;
 x_{i-1} : current position of node $i - 1$;
 x_{i+1} : current position of node $i + 1$;
 E_i : residual energy of node i ;
 E_{i-1} : residual energy of node $i - 1$;
 E_{i+1} : residual energy of node $i + 1$;
repeat
 send x_i and E_i to neighbours $i - 1$ and $i + 1$;
 receive x_{i-1} , x_{i+1} , E_{i-1} and E_{i+1} ;
 set $\lambda(i)_{fw} = \frac{\sqrt{E_i}}{\sqrt{E_i} + \sqrt{E_{i-1}}}$;
 set $position(i)_{fw} = (\lambda(i)_{fw}x_{i-1} + (1 - \lambda(i)_{fw})x_{i+1})$;
 set $\lambda(i)_{bw} = \frac{\sqrt{E_i}}{\sqrt{E_i} + \sqrt{E_{i+1}}}$;
 set $position(i)_{bw} = (\lambda(i)_{bw}x_{i+1} + (1 - \lambda(i)_{bw})x_{i-1})$;
 set $x'_i = (position(i)_{fw} + position(i)_{bw})/2$;
 set $x_i = x'_i$;
until (convergence)
move to x_i ;

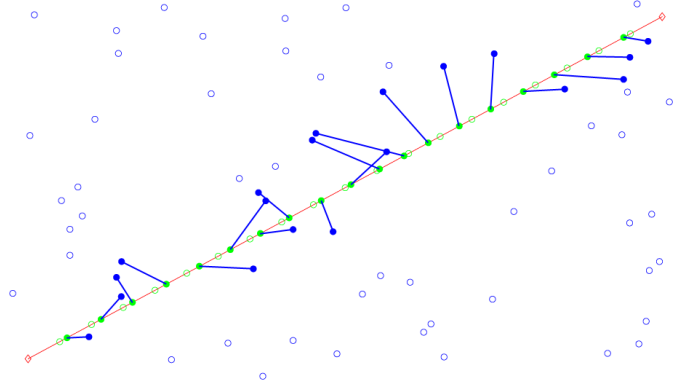


Fig. 3.11. Tracking of nodes movement for the energy spaced virtual movement scheme

In what follows, we summarize the advantages introduced by the virtualization of movements:

- *Travelled distances*: when nodes move all at once, and not iteration by iteration, they can select the most advantageous path to travel in order to reach the final placement. This can all be done with a distributed approach by using local information.
- *Iterations number*: a damping factor, generally, increases the number of iterations needed for an algorithm to converge. Obviously, since the virtual movement does not need to be damped by any factor, we assume that the

number of iterations required for the convergence is the same as for $g = 1$, which is the smallest, as shown in Fig. 3.8 (c). Hence, the convergence is guaranteed to be the quickest.

- *Protocol overhead:* for each iteration, nodes are called to compute their new position and communicate it to their neighbours. Thus, we require that the number of iterations is the minimum, in order not to increase the protocol overhead because of the messages' exchange.
- *Movement's time cost:* we have not designed yet a model that takes into account accelerations and velocities of the nodes, when they move towards their new positions, but it is evident that many intermediate movements are more time consuming than one single movement. Thus, not only does our proposal converge to the algorithmic solution in a smaller number of iterations, but even the execution of the solution is faster than the other algorithm in literature.
- *Movement's energy cost:* the energy model used for the movement is a simple distance proportional cost model, valid for wheeled devices: $E = k \cdot d$, where k is a constant movement [J/m] and d is the travelled distance. This model considers the dynamic friction but it should be enriched by taking into account also the static friction, which represents the resistance to be overcome by a device, in order to start moving: $E = k \cdot d + E_s$, where E_s is the energy needed to overcome the static friction [J]. If the iterations' number increases, also the energy needed for overcoming the static friction will increase. With our scheme, the node moves only once and the term E_s will contribute only once to the total energy expenditure, and for this reason it can be neglected.

Besides these additional advantages, we can conclude that our scheme is: distributed, adaptive, quickly convergent and suitable to be used in self-organizing systems. In respect of the features listed in Section 3.2.4, we need only to show with better proofs, that it is also energy efficient. This will be done in the next section.

Complexity analysis

The complexity of the algorithms "Real Movement" and "Virtual Movement" is closely related to the computation complexity of the "repeat-until" loop. This analysis of complexity should be performed considering best and worst cases that, in our problem, are related to initial positions of the nodes respect to the straight line between two terminal nodes. Rather than a real complexity analysis in this paragraph we focus on the improvement performed by our "Virtual Movement" scheme compared to using damping factor. The mean difference between the two schemes stay in the operation of nodes movement which in "Virtual Movement"'s case is outside of the loop with considerable benefit for the algorithm's convergence. In fact we'll show such a operation, if put within the loop, it increases artificially the number of iterations needed

for the convergence, and then, the time complexity and communication complexity will be higher.

Let the equation of the nodes movements towards their final position:

$$v'_i = v_i + g(v'_i - v_i) = (1 - g)v_i + gv'_i \quad (3.30)$$

we assume, without loss of generality, that nodes previous and successive are fixed.

The final positions due to damped movements of the nodes will be achieved through little steps. These steps by a mathematical point of view can be seen as part of a sum that step by step arrive to final position. Accordingly we can write the following succession of partial sums, each for one algorithm's iteration:

$$\begin{aligned} S_1 &= (1 - g)v_i + gv'_i \\ S_2 &= (1 - g)S_1 + gv'_i \\ S_3 &= (1 - g)S_2 + gv'_i \\ &\dots \\ S_{n+1} &= (1 - g)S_n + gv'_i \end{aligned} \quad (3.31)$$

By explicit the previous equations according to the positions v_i and v'_i we obtain:

$$\begin{aligned} S_1 &= (1 - g)v_i + gv'_i \\ S_2 &= (1 - g)^2v_i + g(1 - g)v'_i + gv'_i \\ S_3 &= (1 - g)^3v_i + g(1 - g)^2v'_i + g(1 - g)v'_i + gv'_i \\ &\dots \\ S_n &= (1 - g)^n v_i + \sum_{k=1}^n g(1 - g)^{n-k} v'_i \end{aligned} \quad (3.32)$$

Before understand how many iterations n are needed for the algorithm's convergence when the damping factor g varies, we make some observations:

1. $(1 - g)^n + \sum_{k=1}^n g \cdot (1 - g)^{n-k} = 1$
2. $\lim_{n \rightarrow \infty} (1 - g)^n = 0$
3. $\lim_{n \rightarrow \infty} \sum_{k=1}^n g(1 - g)^{n-k} = 1$

In substance the coefficients $(1 - g)^n$ and $\sum_{k=1}^n g(1 - g)^{n-k}$ can be considered as a weights on position in the previous iteration and final position respectively, this for each node. Then the final position lie on the straight

line between these two positions because we assumed that previous node and successive node are fixed and will be spaced according on these coefficients just computed. Moreover we know that the sum of these coefficients is always equal to one, and their trend is the same in respect to g and n , then we focus only on the analysis of the first one.

The $(1 - g)^n$ coefficient has an exponential trend as showed in Figure 3.12.

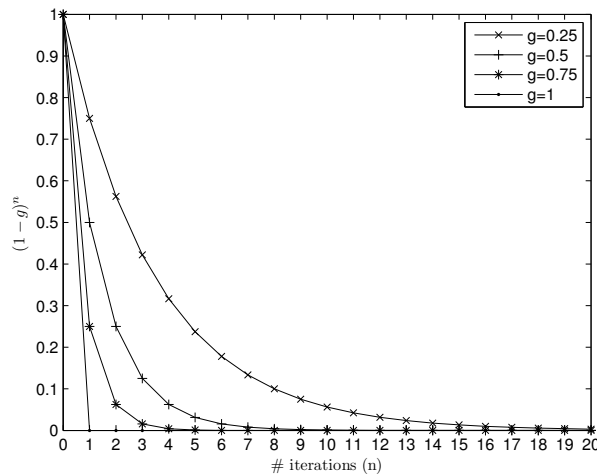


Fig. 3.12. Trend of $(1 - g)^n$ when g and n varies

Each node achieve own final position when $(1 - g)^n = 0$. Figure 3.12 show as this result is obtained for increasing values of n when g decreases. Therefore if we choose a small value of g to smooth the oscillation during the movement phase, the number of iteration to the algorithm's convergence exponentially increases. Finally we remind that virtualizing the movement means choose $g = 1$ that graphically is the case of immediate convergence for $n = 1$.

3.2.5 Results and Discussion

In Sections 3.2.2 and 3.2.2 we showed that the placement of nodes involved in relaying a data flow affects their lifetime significantly. We found out that the choice of the new placement should be driven by the initial residual energy of the nodes. In case this information is not available, we showed that also the evenly spaced placement outperforms the random placement. However, the determination of the best placement requires a central management unit to collect all the needed information and to compute the new positions. In addition to the disadvantages of this centralized approach, if nodes are not provided with controlled mobility, the cost to place them through human

intervention could be very high and often impossible due to environmental limitations. Therefore, we have assumed that nodes are able to communicate with each other and move according to a new distributed and asynchronous algorithm that drives them to the most energy efficient positions. In this Section we want to motivate better the need for this algorithm by showing some simulative results. The simulation scenario and the energy model are the same presented in Section 3.2.3, whose parameters are in Table 3.3. We carried out two simulation campaigns:

- by varying the nodes density ρ , between 2 and $10 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$ and for a fixed distance source-destination $600\sqrt{2}m$;
- by varying the source-destination distance l , between $200\sqrt{2}$ and $1000\sqrt{2}m$ and for a fixed nodes density $6 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$.

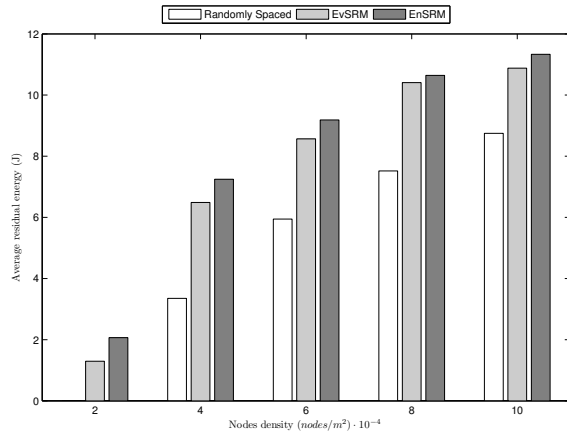
The output parameters that we have considered to evaluate random placement, EvSRM and EnSRM are:

- the average residual energy (and the average energy consumption) of the nodes involved in the data flow, after they moved to the optimal positions and they relayed the data flow;
- the average distance travelled by nodes to reach the optimal positions.

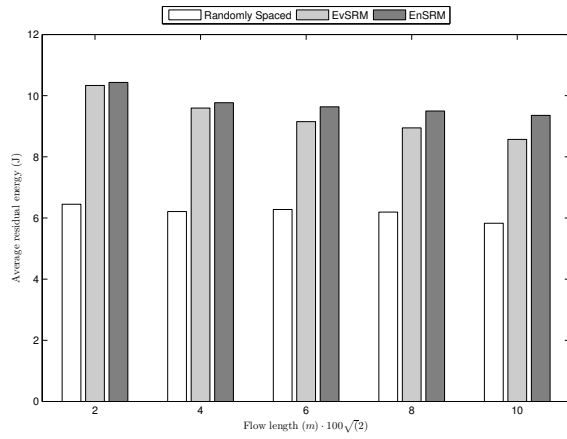
In Fig. 3.13, we can see the average residual energy when the nodes density (a) and the source-destination distance (b) vary. In respect of a similar study we conducted in [92], here we have used a higher value of the parameter P_{rec} ($1.0 \cdot 10^{-11} W/m^2$ instead of $3.16 \cdot 10^{-12} W/m^2$), calculated from [91], in order to give more importance to the correct transmission/reception of data.

In fact, for this value of P_{rec} we can see from Fig. 3.13a and 3.13b that the residual energy of nodes in their original random placement is always lower than the performance achieved by both the schemes of movement and relaying. In the case of very low nodes density, the random placement does not guarantee the correct termination of the data relaying, because at least one of the nodes of the path dies before completing the task. For sake of clarity, we mention again that our definition of path lifetime is taken from [94]. The same observation that we made in Section 3.2.2 for the bidirectional placement computation still applies in this case: we are implicitly assuming to have a perfect bidirectional flow, this assumption contributes to improve the performance of "symmetric" solutions as the EvSRM algorithm. However, even when energy for the movement is taken into consideration, the EnSRM algorithm keeps a margin of improvement in respect of the EvSRM for all the nodes densities, from 3% up to 30%, as in Fig. 3.13a, and this margin increases when the source-destination distance increases, from 2% up to 10%, as in Fig. 3.13b.

The improvements achieved by EnSRM are explained in Fig. 3.14a and 3.14b. In Fig. 3.14a we can see that EnSRM makes nodes move 7% less than EvSRM for all the nodes densities. A higher nodes density means a shorter transmission radius (from Table 3.3), which, in turn, increases the number of

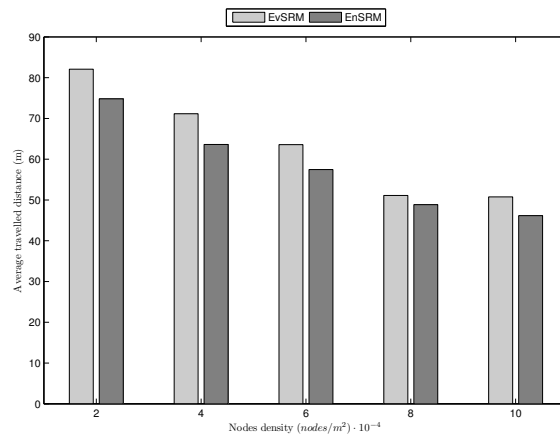


(a)

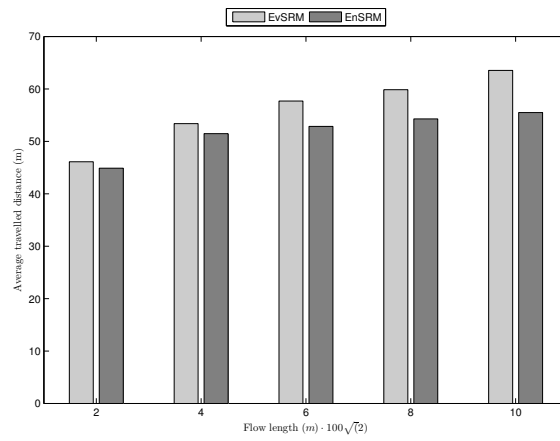


(b)

Fig. 3.13. Nodes average residual energy when a) nodes density, b) source-destination distance vary.



(a)



(b)

Fig. 3.14. Nodes average travelled distance when a) nodes density, b) source-destination distance vary.

nodes involved in the data relaying. A higher number of nodes for the same source-destination distance decreases the average distance travelled by each node. In fact we observe a decreasing trend for both the algorithms when nodes density increases. When the source-destination distance increases (Fig. 3.14b), the number of nodes involved in the data relaying increases proportionally, since the nodes density is fixed. Consequently, the probability for the routing protocol to choose all nodes close to the straight line slightly decreases. Thus, the average distance travelled by nodes should increase. In fact, this is the behaviour of both EvSRM and EnSRM. If we take into consideration the difference between the shortest and the longest source-destination distance, the former requires nodes to move 30% more, while the latter only 16% more. Hence, the gain of EnSRM over EvSRM increases from 4% for the shortest distance up to 16% for the longest distance. If we consider that the routing protocol chooses the nodes that are initially located in the positions closest to the evenly spaced positions on the straight line between source and destination, then we can affirm that the EnSRM makes nodes move, iteration by iteration, along more energy-efficient trajectories.

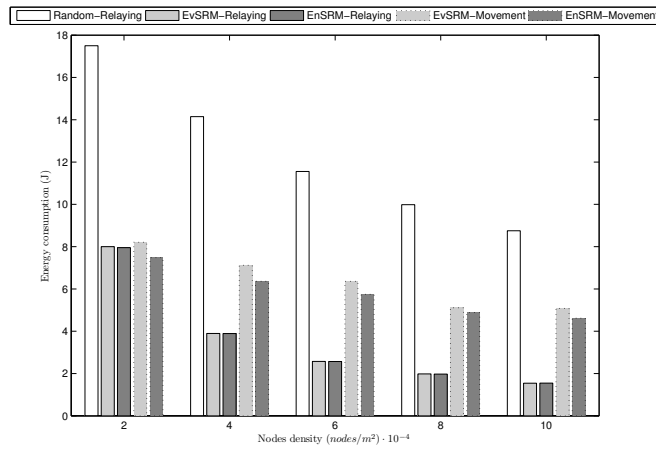
Considering the energy models used for relaying data and movement, we are aware that the largest energy expenditure is due to the movement. In order to quantify this cost, in Fig. 3.15a and 3.15b we plot the average energy consumption for the two actions separately. These plots are complementary to those in Fig. 3.13, but they provide us with some insights into the two sources of energy consumption for the nodes. In Fig. 3.15a we can see that, for both the algorithms, when nodes density is low, the energy consumption for relaying data is very similar to that for moving nodes, but the latter exceeds the former by almost three times when nodes density is high. In Fig. 3.15b we can see that the source-destination distance does not have any impact on the energy used for relaying data. However, it affects the energy used for the movement, as we knew from the travelled distances in Fig. 3.14.

In conclusion, as we already saw in Fig. 3.13, it is always convenient, in terms of energy consumption, to let nodes move and find the optimal placement by any movement algorithm. From Fig. 3.15 we get a more precise idea on the direction to choose in order to improve the algorithms.

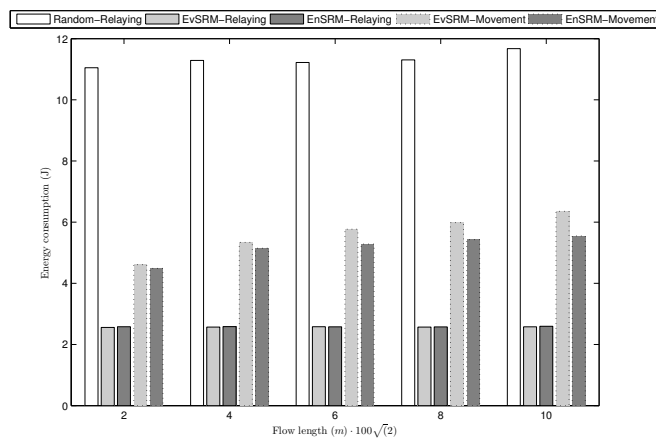
Performance Evaluation

In this last section, we show the results of the Energy Spaced Virtual Movements (EnSVM) and the Evenly Spaced Virtual Movement (EvSVM) schemes, in comparison with the schemes with real movements, in terms of residual energy and travelled distances.

In addition to the assumptions reported in Section 3.2.4 about the evaluation environment, we give here the details and the initial values of the simulation's variables in Table 3.4. The reported value of the damping factor is only used for the schemes without virtualization. We study the performance of the proposed mechanisms by investigating scenarios with variable nodes



(a)



(b)

Fig. 3.15. Average energy consumption when a) nodes density, b) source-destination distance vary.

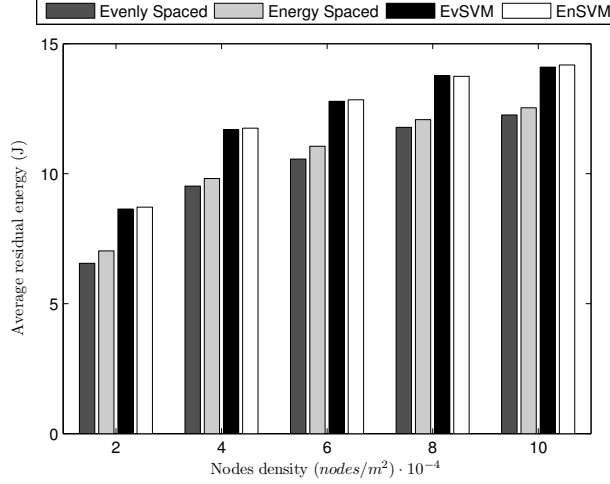


Fig. 3.16. Average residual energy for increasing nodes density

density, ρ , and flow length, l . For all the scenarios, the simulations have been statistically validated by averaging the results over 100 runs, which allows us to reach the wanted interval of confidence (95%).

The energy required to send one bit at the distance d is $E = \beta d^\alpha$, where α is the exponent of the path loss ($2 \leq \alpha \leq 6$), β is a constant [$J/(\text{bits} \cdot \text{m}^\alpha)$]. For α and β we used values typical of the free space model [91].

As we have seen in Table 3.4, the nodes density is considered variable and the schemes performance has been evaluated for 5 different number of nodes: 200, 400, 600, 800 and 1000 in the $1000\text{m} \times 1000\text{m}$ sensor field. The flow length is set to $600\sqrt{2}\text{m}$. With this campaign, we want to study how the number of nodes affects the behaviour of the network, when nodes can move.

Table 3.4. Evaluation Parameters

Field Area ($L \times L$)	$1000\text{m} \times 1000\text{m}$
Nodes Density (ρ)	$[2 \div 10] \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$
Flow Time Length (T_F)	$87.6 \cdot 10^3 \text{h}$
Flow Length (l)	$[200 \div 1000] \sqrt{2}\text{m}$
Maximum Transmission Radius (r)	$1/(2\sqrt{\rho}) \text{m}$
Relay Nodes Number (N)	l/r
Initial Residual Energy Range (E_i)	$15 \div 20 \text{J}$
Minimum Required Power (P_{rec})	$3.16 \cdot 10^{-12} \text{W}/\text{m}^2$
Transmission Rate (r_T)	$1 \text{kb}/\text{s}$
Movement Constant (k)	$0.1 \text{J}/\text{m}$
Damping Factor (g)	0.75
Number of runs for each scenario	100
Statistical confidence interval	95%

Fig. 3.16 shows the average residual energy of the nodes involved in the data flow, after they moved and they relayed the data for the given flow time length, T_F . As we can see the level of energy of the virtual schemes is always larger than the schemes with real movements. In respect of the Evenly Spaced scheme, it improves the performance by about 15-32%, while for the Energy Spaced that is able to take better advantage of higher densities, we have a decreasing improvement that varies between 23% and 13%.

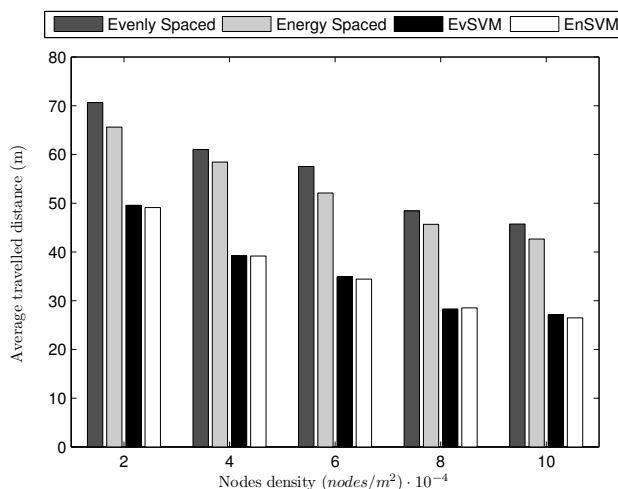


Fig. 3.17. Average distance travelled by nodes for increasing nodes density

We have to recall that in the energy model for the movement there is no consideration of the energy needed to overcome the static friction, which would decrease even more the residual energy of the schemes without virtualization of the movements. As we can see, both EvSVM and EnSVM achieve similar performances, because they both save the energy for the movement, which is the highest part in the total energy consumption.

In fact, in Fig. 3.17 we see that the distances travelled by the schemes with virtualization of the movements are from 25% up to 42% less than the distances travelled by the schemes without virtualization. Besides the direct effect on the residual energy, this means also that there is a smaller possibility of damaging or consuming the mechanical devices used for the movement.

In the second simulation campaign, we want to show the algorithms' performance when the physical distance between the two static terminal nodes varies between $200\sqrt{2}m$ and $1000\sqrt{2}m$, with an increase step of $200\sqrt{2}m$, while the nodes density is set to $6 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$. The upper limit is given by the size of the sensor field and it represents the case that the fixed terminals are placed at the two furthest corners of the field. This campaign is useful to

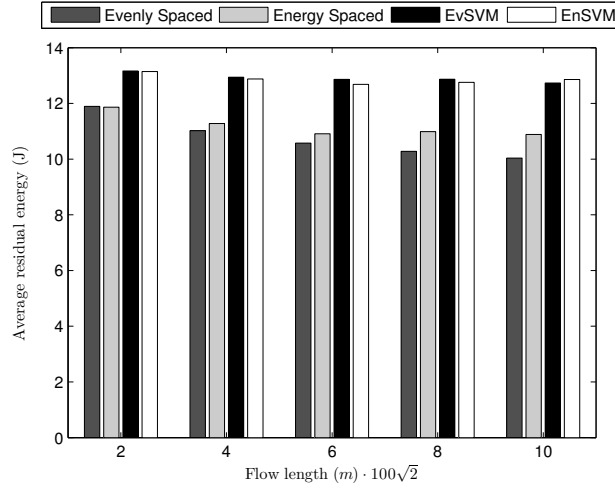


Fig. 3.18. Average residual energy for increasing flow length

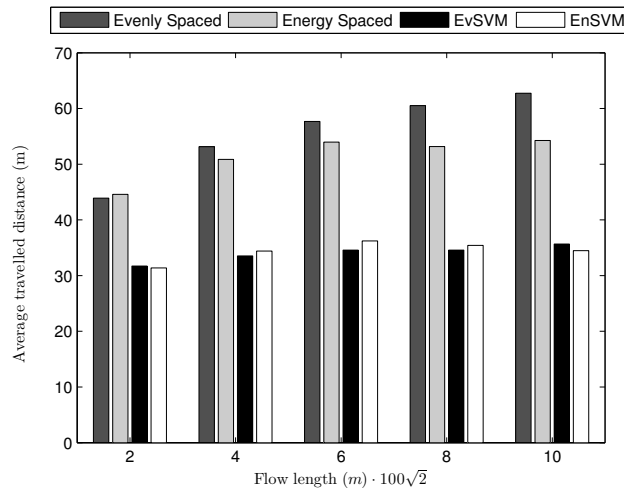


Fig. 3.19. Average distance travelled by nodes for increasing flow length

understand if longer flows make nodes use more energy for travelling longer distances, as it was for the schemes with real movement. As we can see in both Fig. 3.18 and 3.19, when the movement is virtual, residual energy and travelled distance are not affected by the length of the flow, in fact they are quite constant for all the values of flow length.

In Fig. 3.18 we can see that, for this scenario, the improvement reported by the virtual movement schemes is 10%÷20% and 9%÷18% in respect of

evenly spaced and energy spaced with real movement, respectively. Fig. 3.19 shows that, even for this scenario, the virtual movement schemes make the nodes move from 28% to 45% and from 29% to 36% less than the distances travelled by the evenly spaced and the energy spaced scheme, respectively. For our simulations the damping factor is set to 0.75, but, when the damping factor goes down to 0.25, the results would still be remarkable: 20%÷30% less of the distances travelled by the schemes with real movement (see as reference also Fig. 3.8 (b)).

Conclusion

In this work we considered the concept of controlled mobility in a novel fashion. Specifically, controlled mobility is used as a new design dimension of the WSN to move nodes towards the best placement for prolonging the lifetime and for maximizing the energy efficiency. In this section we developed a distributed algorithm based on virtual movement of nodes. The movement virtualization is a smart and very simple approach for exploiting the advantages of better nodes placement, without wasting energy because of the mobility. This new mechanism exhibits all the important features that a WSN placement and movement algorithm should possess: it is distributed, adaptive, quickly convergent and suitable to be used in self-organizing systems. Furthermore, it has been deeply analyzed through simulations, and it has been compared with other existing schemes that use real movement. In all the simulation scenarios, our mechanism outperforms the existing mechanisms in terms of energy consumption and travelled distances.

3.3 Controlled mobility assisted Routing protocol

There are many challenges to face while designing wireless networks and protocols, such as obtaining a good throughput, minimizing data delay, minimizing energy waste, etc. In fact, most of the wireless networks are characterized by battery-equipped devices, thus the minimization of the energy consumption is a key factor. With the miniaturization of computing elements, we have seen many mobile devices appear in the market that can collaborate in an ad hoc fashion without requiring any previous infrastructure control. This gave birth to the concept of self-organization for wireless networks, which is intrinsically tied to the capability of the nodes to move to different placements. In the last few years, the research community has become interested in the synergic effect of mobility and wireless networks. Controlled mobility is a new concept for telecommunication research field and can be defined as a kind of mobility where mobile devices are introduced in the network and move to specified destinations with defined mobility patterns for specific objectives. In practice, controlled mobility is a new design and control primitive for the network.

The use of controlled mobility as a new design feature to enhance the performance of wireless networks represents a recent, innovative and revolutionary concept. In fact, while opportunistic use of external mobility has been extensively investigated, the use of controlled mobility is largely unexplored. This new design dimension can effectively be used to improve system performance by allowing devices equipped with mobility support to reach favourable locations. Although many communication protocols for wireless networks have been proposed, to the best of our knowledge there is no routing protocol based on controlled mobility. In fact, in [102] authors consider jointly mobility and routing algorithms but the solution they propose is for Wireless Sensor Networks (WSNs) and only the base station is mobile. Among the main routing protocols proposed for Mobile Ad hoc NETWORKS (MANETs), we have the Ad hoc On Demand Distance Vector (AODV) routing [103] and [104] and the Temporally Ordered Routing Algorithm (TORA) [105]. Both are examples of demand-driven algorithms that eliminate most of the overhead associated with table update in high-mobility scenarios. However, the path discovery phase incurs in high energy costs. On the other hand, our system is quasi-static, in the sense that the only mobility we consider is controlled mobility, which is used by nodes to reach specific locations, then, for energy efficiency matters, it is convenient to use a table-driven system. Another interesting routing protocol is [101], where the minimum metric paths are based on two different power metrics:

- Minimum energy per packet
- Minimum cost per packet

However, this routing algorithm does not take into account the mobility as a new design dimension. In this section, we take the controlled mobility into account by investigating the performance of a wireless network, where all the devices are equipped with mobility unit. The idea is to use existing multi-hop routing protocol, specifically we consider the well-known routing algorithm AODV, and achieve further improvements in terms of network performance as throughput, data delay and energy spent per packet, by explicitly exploiting mobility capabilities of the wireless devices. Previous analytical results formulated in [106], [107] and [108] suggests that controlled mobility of nodes helps to improve network performance. Based on these results, we consider jointly controlled mobility and routing strategies. We perform simulations through a well-known simulation tool [109] to quantify the throughput, delay and energy spent per packet compared with wireless network where AODV is used. This work is presented in [2].

3.3.1 State of art

In the recent past a lot of works studied the effects of mobility in the networks. Often, devices' mobility has been regarded as a negative fact that

causes link break, disconnections, etc. From a certain moment it has been understood that mobility of nodes can potentially be used to improve performance of the network. Grossglauser and Tse [110] showed that mobility increases capacity of a network. Unfortunately, they did not take into account the delay in their work. The research community investigated thoroughly the delay-throughput trade-off and some interesting results have been obtained. In fact, Gamal [111] determined the throughput-delay trade-off in a fixed and mobile ad hoc network. He showed that, for n nodes, the following statement holds $D(n) = \Omega(nT(n))$, where $D(n)$ and $T(n)$ are the delay and the throughput, respectively. For a network consisting of mobile nodes, he showed that the delay scales as $\Omega(n^{1/2}/v(n))$, where $v(n)$ is the velocity of the mobile nodes. Once the trade-off between delay and throughput has been characterized, some algorithms that attain the optimal delay for each throughput value, have been proposed. Another model makes it possible to exploit the random waypoint mobility of some nodes, in order to design a routing algorithm that allows high throughput with low delays, where the delay depends on the nodes' mobility, while the throughput is independent of it [112]. De Moraes [113] showed that there is a trade-off among mobility, capacity and delay in ad hoc networks. A first step in taking advantage of the possibilities that mobility introduces has been made by the research community when predictable mobility became an important research focus. In fact, researchers studied many specific network objectives, under a random mobility-based communication paradigm, nevertheless the mobility of the sinks, for example in military applications, is based on soldier or fire fighter movements, thus, it is predictable, in substance. Generally, the existing research in wireless sensor networks considers sink movement based on random mobility. However, the trajectories of the sink, in many practical applications, can be determined in advance. Based on these considerations, Lee [114] proposed a predictable mobility-based algorithm, which uses the existing dissemination protocols and it is based on the random mobility-based communication paradigm. He showed the improvements and the various advantages of using the predictable mobility-based communication paradigm as the energy consumption decreases and the network lifetime increases. Predictable mobility of nodes has also been exploited to help in packets delivering [115]. In this work, nodes routing tables are updated with link state and trajectory information, which are received from other nodes. The problem of routing related to the predictable mobility has also been analyzed by [116]. In this work, paths are created by the movements of nodes, which will deliver the message they are carrying when they find other suitable nodes. The space-time routing framework it proposed leverages the predictability of nodes motion. Controlled mobility has been a hot research topic of the robotics community for many years. It concerns the motion coordination of a group of robots for a common objective, typically the coverage of a geographical area. In [117], the authors consider the problem of deploying a mobile sensors network composed of a distributed collection of nodes equipped with locomotion capability. Such mobile nodes use their ability to move in order to maximize the area covered

by the network. Their approach is based on a potential-field approach and nodes are treated as virtual particles, subject to virtual forces. The concept of controlled mobility is also used by [118] by considering a hybrid network with both static and mobile nodes, which fully exploits the movement capability of the sensors. In [102] authors consider jointly mobility and routing algorithms, but the solution they proposed is based on the base station as the only controlled mobile device. In this work we are interested to consider the mobility of devices in a controlled fashion along with the routing algorithm. Specifically, we base our proposal on the analytical results obtained in [106] and [108] that show the potential advantages obtainable through controlled mobility. In [108], it was not possible to take into account all the constraints of a real routing algorithm and, for this reason, we implemented RPCM in a well-known simulation tool, ns2.

3.3.2 Practical Applications of Controlled Mobility

In this section we will give some practical applications of Controlled Mobility, which covers several research areas such as robotics, software engineering, optimization, swarm intelligence, etc. There is an interesting real application of controlled mobility to reduce power consumption realized by Intel, that showed as a few motes equipped with 802.11 wireless capabilities can be added to a sensor network in order to act as wireless hubs [98]. Load balancing through controlled mobility in wireless sensor networks is studied by Luo and Hubaux in [15]. The nodes closest to the base station are the bottleneck in the forwarding of data. A base station, which moves according to an arbitrary trajectory, continuously changes the closest nodes and solves the problem. The authors find the best mobility pattern for the base station in order to ensure an even balance of network load on the nodes. Recently, Intel installed a small sensor network in a vineyard in Oregon and a second one in Northern California to monitor microclimates and Redwood canopies, respectively. In this context, the mobile sensors had to measure, share and combine the collected data regarding temperature, humidity, and other factors. At the gateway, the data was interpreted and used to help avoid mold, frostbite, and other agricultural problems. The agricultural environment is just an example of how a sensor network can take advantage of mobile robot's capabilities for data gathering and interpretation. Furthermore, sensors often need to be recalibrated and a robot could act as a gateway to the sensor network and perform calibration tasks. An interesting application of sensor devices with mobile robots, related to coverage, is for people with disabilities [99] and [100]. In fact, technical devices, such as mobile robots can aid personal assistance. A mobile robot requires a sensing system in order to control the path of movement and the surrounding environment. The robot can be equipped with sensors for detecting distances and obstacles. Another work worth mentioning has been conducted by Kansal in [19]. The authors do not present an algorithm for the optimization of some parameters, instead they

propose Morph as a new vision of sensor networking, where controlled mobility is considered as an additional design dimension of the communication protocols. They argue that, in Morph, controlled mobility can be employed for the sustainability of the network, which consists in both alleviating the lack of resources and improving the network performance.

3.3.3 A new Routing Protocol based on Controlled Mobility (RPCM)

The network scenario we consider consists of all nodes able to move and control their movements. The communication strategy used in this work considers different paths for each pair source-destination nodes and the best path is selected to be used for data communication. The choice of the best path is based on a metric. Specifically, in this context we consider the path whose nodes have to travel the total minimum distance to reach the evenly spaced positions on the straight line between the source-destination pair. The same metric has been proposed in an optimization model in [108], where the model determines the placement that minimizes the total travelled distance of the sensor nodes. This kind of movement could be useful in all situations where a mobility too high can be dangerous (i.e. military applications such as minefields monitoring) or difficult because of a high presence of obstacles.

Some Assumptions

Assume n nodes deployed randomly in a square area. All the nodes have the same transmission range. If two nodes are within each other's transmission range, they can communicate directly and they are *neighbours*. Otherwise they have to rely on intermediate nodes to relay the messages for them. Any node in the network may have data to be sent to any other node. The path from a source to a destination may not be direct but involves other intermediate nodes. We assume that several paths, from the source s to the destination d , have already been discovered by a routing protocol. Specifically, we apply the Route Request phase of the AODV protocol, with some additional information such as the nodes' position. We need this information for implementing our protocol as explained in the follow. We also assume nodes move only in order to reach specific locations when they belong to a path. Our mobility control scheme is not directly incorporated into a specific layer of the classical ISO/OSI layer structure (i.e. physical layer, routing layer, etc.), because mobility management is transversal to all the layers and controlled mobility can be exploited by any of the layers, as we can observe in Fig. 3.20. In fact, controlled mobility could be exploited at different levels.

The Routing Protocol based on Controlled Mobility

In this section we detail the routing protocol along with the mobility algorithm. Assume a source node s needs to establish a communication with a node

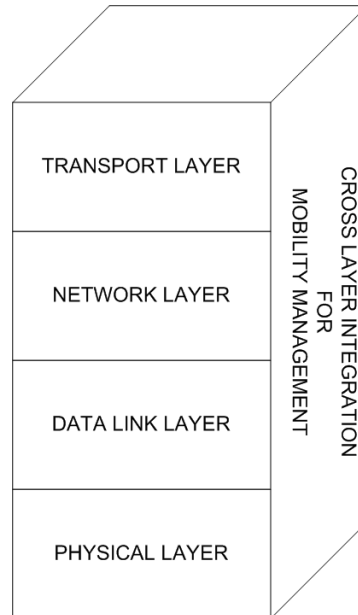


Fig. 3.20. Protocols' stack integrated with mobility management.

d. The source node will broadcast a Request Packet, which will be forwarded by its neighbours. Each node includes in the request packet its geographical coordinates in the network. Once the Request Packet reaches the correct destination, the node d will not send a Reply Packet immediately, but it will wait for processing other requests. In order to avoid unuseful delay, a destination node, will wait for a specific time, and then it will send a reply packet by building the best selected path. The metric we introduced to evaluate the goodness of a path is based on the total travelling distance. In practice, the algorithm will choose the path that minimizes the sum of nodes' travelled distances. Other metrics, such as the minimization of the maximum travelled distance, could be considered and implemented. In Fig. 3.21 the request phase of the routing protocol is explained. We can observe that the source node starts a request phase by sending a Route Request and every intermediate node stores the position of the previous node, the ID of the previous node and re-broadcasts the request packet. The mechanisms to avoid loop and control packet storms are the same as in AODV. Once the Request Packet reaches the destination node, if the request is processed for the first time, the destination node d activates a timer and continues to process other Request Packets of the same source node s . Otherwise, d compares the previous path with the current path and selects the best one (in this case the path whose nodes travel the minimum total distance). Once the timer expires, d sends a Reply Packet to the first node of the selected path in the backward direction. This

node computes its new position depending on the number of nodes involved in the path (this information is sent from the d node) and forwards the Reply Packet to the following node in the backward direction (this information has been stored in the Routing Table during the Route Request phase). Hence, this node will move to the evenly spaced position on the straight line between the source and the destination. When each relay node knows its position, the optimal configuration of relay nodes for an active flow is established as in [106] and [5]. It is worth to note that in this case the solutions found in [106] and [5] are the same, because the initial energy of nodes is the same. In practice, the nodes will reach the evenly spaced positions on the straight line between the source and the destination.

In Fig. 3.22 the Reply phase is explained. Once the source node s receives the Reply Packet, all the nodes belonging to the path have already moved to their new position and s will start the data communication flow.

In practice, nodes compute their new positions based on the total number of nodes belonging to the current path and considering they have to be evenly distributed on the straight line between the source and the destination. The Mobility Algorithm can be summarized as follows:

Mobility Algorithm: Mobility control at each relay node.

- Each node knows the position of the source from the request packet, and acquires the information regarding the position of the destination, the number of nodes involved and its position in the path from the reply packet. From the source and destination positions, it calculates the distance between the two terminal nodes. From the number of hops and its position in the path, it is able to compute its new evenly spaced position between source and destination;
- Each node that received a reply packet moves towards its new position.

In [5], authors had to introduce a damping factor g to avoid oscillations in the network. In fact, nodes exchange local position information with neighbours and some iteration of the distributed algorithm is needed to reach the final optimal displacement of the nodes. Thus, we do not need to introduce any damping factor, because nodes already have all the information they need to reach the new location. Furthermore, we do not have any convergence concern. In fact, nodes start to move once they receive the reply packet and reach the final destination.

From the description of the routing protocol is clear the reason why no damping factor is needed, even if our protocol is totally distributed and the Mobility Control Algorithm is orthogonal to the network layer. The new protocol requires few changes to the classical schemes, the information that need to be added are:

- in the Request Packet: the positions and the IDs of source and forwarding nodes;
- in the Reply Packet: the positions of the destination node and the hop number of the source-destination path;

- in the nodes' routing tables: source's position and hop number of the source-destination path.

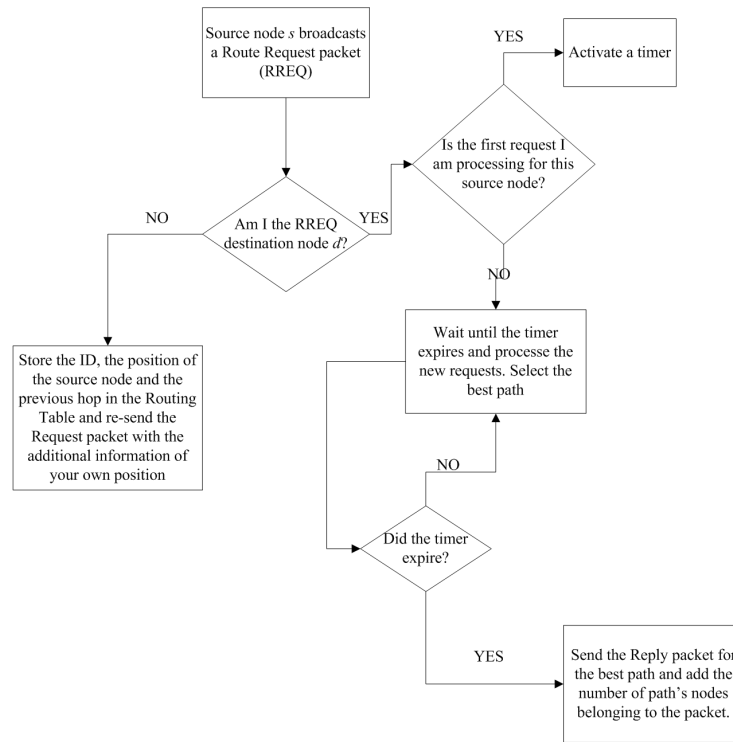


Fig. 3.21. Routing Request phase of RPCM.

The effect of applying the RPCM protocol is shown in Fig.3.23 and Fig.3.24.

In Fig.3.23 we can observe the selection of many potential paths for the pair source-destination, node 0 and 99, respectively. Specifically, the paths discovered are: 0-53-89-72-38-99, 0-77-15-8-98-99, 0-16-70-71-19-99, 0-12-9-74-97-99 and 0-32-17-38-99. Among the different paths, the one whose nodes travel the total minimum distance is chosen. In this case the selected path based on our metric is 0-32-17-38-99. When the reply phase begins, the first node that receives the reply packet is node 38, it computes its new location and sends the reply packet to the node 17, then it moves towards its best location. In similar fashion, node 17 receives the reply packet from the node 38, computes its new destination, sends the reply packet and moves to the new location, etc. Once node 0 receives the reply packet from the node 32, then all the nodes belonging to the path (in this case, nodes 38, 17 and 32) already

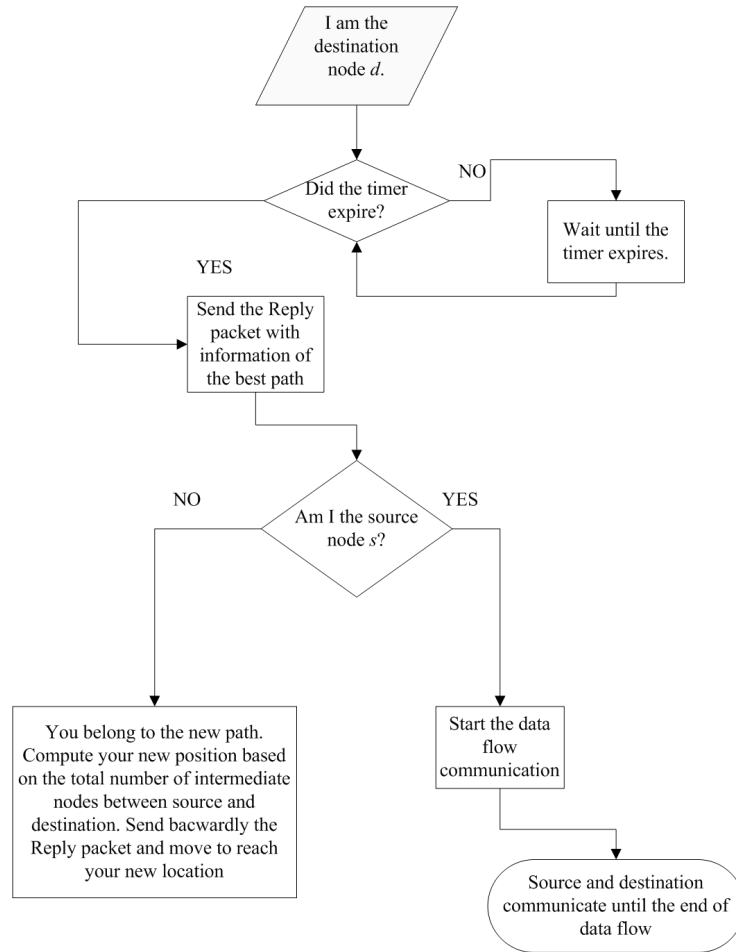


Fig. 3.22. Reply phase of RPCM.

moved to their new positions and node 0 starts the data communication flow to node 99. Note that source node and destination do not move.

3.3.4 Simulation and Results

As we already said in the previous subsections, the optimization model including the minimization of the nodes' total travelled distances along with other possible metrics has been introduced in [108]. Unfortunately in that analytical work, many practical details could not be taken into consideration. For this reason, we chose to implement one of those possible metrics in a complete routing algorithm and simulate its behaviour in a well-known network simu-

lator: ns2, in order to evaluate the realistic effects of controlled mobility in the routing process in comparison with the AODV protocol.

Reference Environment

In Table 3.5, all the most important environment and simulator parameters are reported. We chose to implement our algorithm in a square area of $500m \times 500m$, where a variable number of wireless nodes has been randomly deployed, according to the reported nodes density (2, 3, 4, 5, $6 \frac{nodes}{m^2}$, which correspond to 50, 75, 100, 125, 150 nodes). Also the number of concurrent flows is considered variable. Depending on the density, nodes have a different transmission area to cover. All nodes have initially the same energy and transmit at the same transmission rate, when they move the energy expenditure E_M is proportional to the travelled distance d by a movement factor k . When nodes mobility is allowed, the set of limitations becomes enriched with new elements. In fact, the definitions of an energy model related with the motion of nodes and of another model related with the communication needed for their coordination are required. For the former a simplified model is a distance proportional model $E_M(d) = kd + \gamma$, where d is the distance to cover, $k[J/m]$ takes into account the kinetic friction, while $\gamma[J]$ represents the energy necessary to win the static friction, both these constants depend on the environment (harsh or smooth ground, air, surface or deep water). For the latter, usually the energy required to send one bit at the distance d is $E_C(d) = \beta d^\alpha$, where α is the exponent of the path loss ($2 \leq \alpha \leq 4$) depending on the environment and β is a constant [$J/(\text{bits } m^\alpha)$]. In this work we assumed a smooth ground and we fixed the values of the constant $k = 0.1$ and the constant $\gamma = 0$, that are typical values used in simulations. Regarding the simulator, we used a two-ray ground propagation model and both the simulated routing protocols (AODV and RPCM) are mounted on top of the IEEE 802.11 MAC. The energy spent in sleep, wake-up and active mode are reported in the table. The output parameters taken in consideration are:

- throughput,
- delay,
- energy spent for received packet.

We ran 10 simulations for each scenario and average the results in order to reach the 95% of statistical confidence.

Performance Evaluation

We performed two simulation campaigns: the first consists of increasing nodes density for a fixed number of flows ($f = 6$), in the second the number of flows varies between 4 and 12 and the nodes density is set to $\rho = 4 \frac{nodes}{m^2}$. Figures 3.25, 3.26 and 3.27 show the performance of the two algorithms for

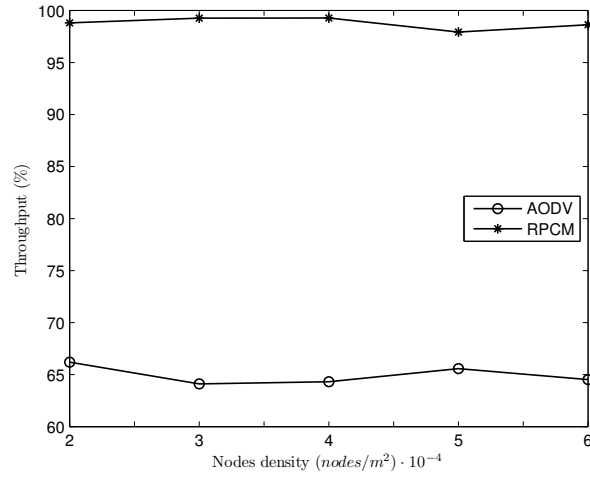


Fig. 3.25. Performance of AODV and RPCM in terms of throughput, when $f = 6$.

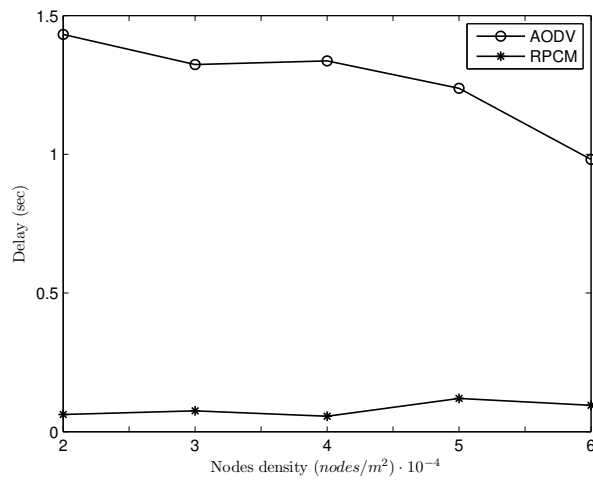


Fig. 3.26. Performance of AODV and RPCM in terms of delay, when $f = 6$.

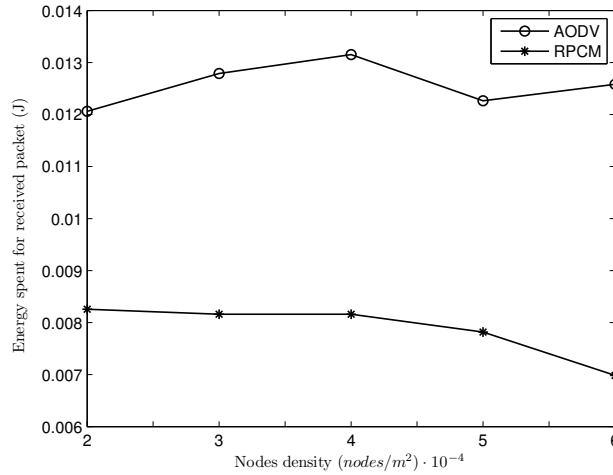


Fig. 3.27. Performance of AODV and RPCM in terms of energy spent for received packet, when $f = 6$.

the first simulation campaign in terms of throughput, delay and energy spent for received packed, respectively.

As we can see, for all the output parameters our scheme outperforms the AODV achieving 30%, 80% and 40% of improvements for throughput, delay and energy spent for received packet, respectively. Furthermore, the behaviour of the RPCM scheme is more robust and scalable than the AODV, since it is almost constant for all the output parameters when density increases, while, in the AODV scheme, delay and energy spent are affected by the nodes density.

Table 3.5. Evaluation Parameters

Environment	
Field Area ($L \times L$)	500m x 500m
Nodes Density (ρ)	$[2 \div 6] \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$
Flows Number (f)	[4 ÷ 12]
Maximum Transmission Radius (r)	$1/(2\sqrt{\rho})$ m
Initial Residual Energy Range (E_i)	100 J
Transmission Rate (r_T)	32 kb/s
Movement Constant (k)	0.1 J/m
Simulator	
Propagation Model	Two-Ray Ground
MAC Type	IEEE 802.11
Packet Size	512 byte
set val(rxPower)	0.00175 W
set val(txPower)	0.00175 W
Wake-Up Time	0.005 s
Number of runs	10
Statistical confidence interval	95%

Figures 3.28, 3.29 and 3.30 show the performance of the two algorithms for the second simulation campaign in terms of throughput, delay and energy spent for received packed, respectively.

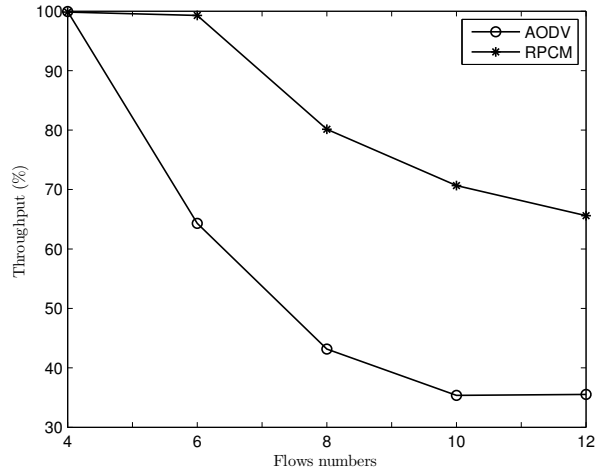


Fig. 3.28. Performance of AODV and RPCM in terms of throughput, when $\rho = 4 \frac{nodes}{m^2}$.

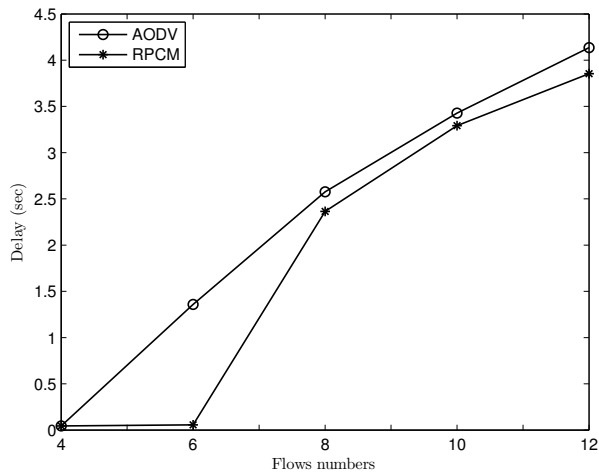


Fig. 3.29. Performance of AODV and RPCM in terms of delay, when $\rho = 4 \frac{nodes}{m^2}$.

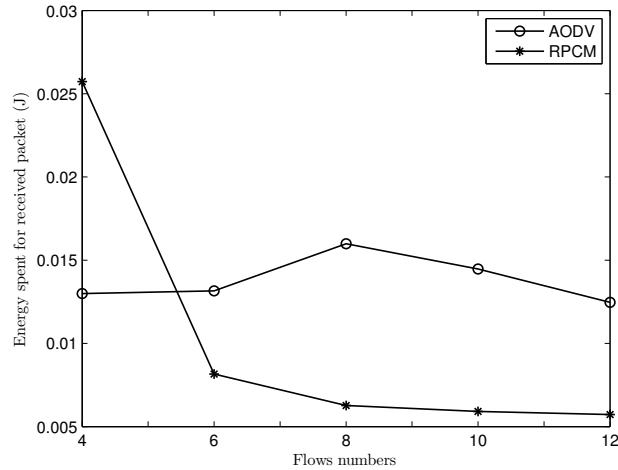


Fig. 3.30. Performance of AODV and RPCM in terms of energy spent for received packet, when $\rho = 4 \frac{\text{nodes}}{\text{m}^2}$.

Also when the number of simultaneous flows varies between 4 and 12, the throughput of RPCM results 30% higher than the AODV on average. The delay is still lower for the RPCM, even if, when the number of flows increases, the improvement tends to reduce. From Fig. 3.28 and 3.29 we see that, for RPCM, we can determine a “threshold” on the number of concurrent flows, until it is below 6, the performance is constant with very high throughputs and very low delays, when the number of flows is higher than 6, then the performance worsens. This result gives the designer a good hint about the number of concurrent flows to allow into the network, in order to have high performance. At last, the energy spent for received packet shows two different trends for AODV and RPCM, the first is not very affected by the number of flows and oscillates between 0.012 and 0.015 J , while the second shows a negative exponential behaviour, for $f = 4$ the energy spent on average is 0.025 J but it reduces till 0.005 J when the number of flows increases.

3.3.5 Conclusion

In this work we focused on both the novel concept of controlled mobility and the routing algorithms. The concept of controlled mobility has been introduced in some previous recent work, but it has only been considered from an analytical point of view or in a marginal fashion, such as only a mobile base station in the network. In this section we focus on the controlled mobility as a new design dimension and we exploit it by implementing a new routing protocol based on controlled mobility. The most important aspect of this is

related to the evaluation performance based on the usage of a well-known simulation tool, ns2. In fact, in previous works the analytical approach limited the use of controlled mobility while in this context, thanks to the simulator, we have been able to consider many realistic aspects of the network, while a routing protocol is implemented. Extensive simulations have been conducted and simulation results have shown how the new routing protocol outperforms a well-known routing algorithm, the AODV. Furthermore, results obtained suggest that other metrics can be easily realized and tested by simulation. In fact, as future works, we intend to study other optimization metrics such as the maximization of the network lifetime or the minimization of the average (or the maximum) distance travelled by nodes belonging to a path.

3.4 A Discrete Stochastic Process for Coverage Analysis

This section considers a network of unmanned aerial vehicles (UAVs). Each UAV is equipped with a certain kind of on-board sensor, for example, a camera or a different sensor, taking snapshots of the ground area. The general aim of the UAV network is to explore a given area, i.e., to somehow “cover” this area using several snapshots. Such a goal is relevant to several applications: target or event detection and tracking in an unknown area; monitoring geographically inaccessible or dangerous areas (e.g., wildfire, volcano), or assisting emergency personnel in case of disasters. Recently, several researchers in the domains of robotics and mobile networking have focused on designing such UAV networks. Research takes place in various areas, e.g., control engineering, communication networking, mission planning, and image processing. A UAV is sometimes also called *drone*.

Our objective is to provide a simple analytical method to evaluate the performance of different UAV mobility patterns in terms of their coverage distribution. To this end, we propose a stochastic model using a Markov chain. The states are the locations of drones, and the transitions are determined by the mobility model of interest. Such a model can easily be created for independent mobility models, such as the random walk and random direction. However, for a cooperative network, in which each drone decides where to move based on the information received from other drones in its communication range, creating a simple Markov model is not straightforward. Therefore, in this work, in addition to providing the necessary transition probabilities for random walk and random direction, we also propose an approximation to these probabilities for a cooperative network. While we choose intuitive rules for the movement paths when two or more drones “meet each other,” the proposed model can be extended such that other rules can be incorporated. We show the validity of the proposed tool by comparing the analytical results with simulations for several scenarios with different network sizes as well as different geographical area sizes. With this tool, steady-state coverage distribution, average and full coverage times for random walk, direction and cooperative mobility models

are evaluated, where the analysis and simulation are in good agreement. This work is presented in [4].

3.4.1 State of art

Several mobility models for autonomous agents have been proposed recently. Some of these are synthetic like the random walk and random direction others are realistic and, all of them, are used mainly to describe the movement of the users in a given environment. In the UAV domain, such models are good for comparison of different approaches, but can give incorrect results when UAVs are performing cooperative tasks [142].

Recently, several research works have shown how mobility can increase throughput [110], energy efficiency [5], coverage [150], and other network parameters. Therefore, the analysis of mobility models has become a highlight to design the mobility of the nodes in a way to improve the network performance. A tool to analyze mobility models is proposed in [149], where the authors model random waypoint-like models as a renewal process to show the steady-state distribution of the speed, while a spatial analysis of different mobility models is provided in [147].

Also, the robotics community is involved in problems related to the coverage of an unknown environment also known as the *sweeping problem* [140]. Basically, the problem can either be solved by providing abilities for localization and map building first or by directly deriving an algorithm that performs sweeping without explicit mapping of the area. In [139], an exploration algorithm that allows multiple robots to cooperatively sweep an area is described. Instead of a measure of coverage, the authors measure the average event detection time for evaluating their algorithm. In addition, coverage problem is sometimes referred to as mapping of an unknown environment and there are useful methods summarized next for motion control and navigation, but they are not directly applicable to coverage or *sweeping* analysis. In [143], the authors introduce the concept of occupancy grid that is a stochastic estimate of the obstacle coverage of the cells obtained by sensing the environment and can be used for both mapping and navigation. Another technique proposed in [148] permits not only the mapping, but also the localization of the robot on the map.

In this section, we focus on the sweeping of an unknown area by probabilistic mobility patterns. Our contribution is to provide an analytical tool to represent existing and possibly new mobility models. We achieve this by providing transition probabilities among positions on a discrete grid and we give a means to compare different mobility patterns in terms of achieved area coverage at a given time or, even better, to design a new model that is able to achieve a desired coverage.

3.4.2 Markov Chain and Coverage Metrics

Markov Chain

We introduce a discrete-time, discrete-value stochastic process that can be used to analyze the coverage performance of a UAV network. Nodes can operate independently or in a cooperative manner. The system area is modeled as a two-dimensional lattice where drones move from one grid point to another in each time step. We assume that a drone can only move to the 4 nearest neighboring grid points (the von Neumann Neighborhood of radius 1 [141]). The probability of moving to a neighboring grid point is determined by the mobility model of interest. In the following, we present the two main components of the proposed Markov chain: state probabilities and transition probabilities.

In our model, the states are defined as $[(\text{Current Location}); (\text{Previous Location})]$ and Fig. 3.31 illustrates the potential states for a 3x3 grid. Depending on the location, the number of associated states is different. Observe from Fig. 3.31 that if the current location is at a corner, boundary, or middle grid point, there are 2, 3, and 4 associated states, respectively. The arrows in the figure represent *potential* transitions between the states.

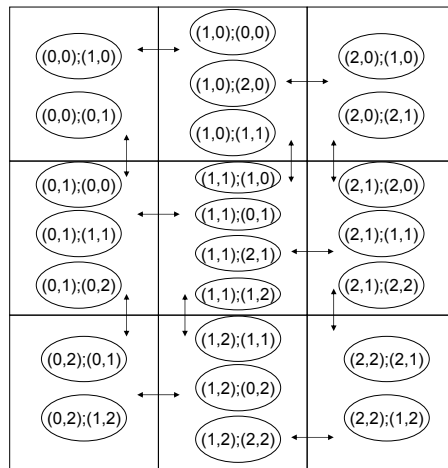


Fig. 3.31. The potential states of the Markov-chain for a 3x3 grid

As an example, Fig. 3.32 shows the state transitions for the state $[(1, 1); (0, 1)]$ in more detail, where P_F , P_B , P_L , and P_R are the probabilities to move forward, backward, left, and right, respectively. Since the previous location is given to be $(0, 1)$, there can be a transition from all 3 associated states of location $(0, 1)$ to $[(1, 1); (0, 1)]$. For this state, the corresponding forward di-

rection from $[(1, 1); (0, 1)]$ is toward $(2, 1)$, then left direction is toward $(1, 0)$, right direction is toward $(1, 2)$, and finally, backward direction is $(0, 1)$.

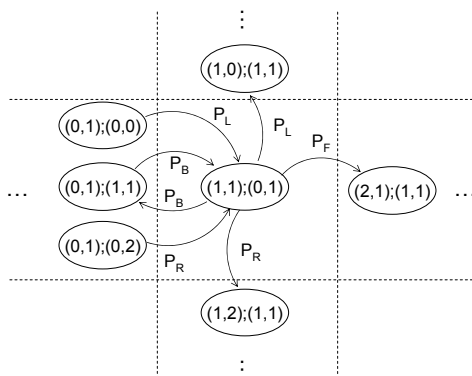


Fig. 3.32. State transition example for state $[(1,1);(0,1)]$

We denote the steady state probabilities of this Markov chain by $\pi = [\pi_{(i,j);(k,l)}]$ and the transition probability matrix by \mathbf{T} , where the entries of the matrix are the transition probabilities between the states $[(i, j); (k, l)]$. Accordingly, we denote the transient state probabilities by $\pi^{(n)} = [\pi_{(i,j);(k,l)}^{(n)}]$, at time step n . Then, we can write the following well-known relations for the steady-state and transient state probabilities [144]:

$$\begin{aligned} \pi &= \pi \mathbf{T} && \text{(for steady-state)} \\ \pi^{(n)} &= \pi^{(0)} \mathbf{T}^n && \text{(for transient state)} \\ \lim_{n \rightarrow \infty} \pi^{(n)} &= \pi && \end{aligned} \tag{3.33}$$

where $\sum \pi_{(i,j);(k,l)} = 1$ and without loss of generalization the initial-state $\pi^{(0)}$ can be chosen to be $[1, 0, \dots, 0]$ (since the solution for π is independent of the initial condition). From these linear equations, we can obtain the steady and transient state probabilities, which will be used to determine the coverage of a given mobility pattern.

Coverage Metrics

We denote the steady state coverage probability distribution for an axa grid by $\mathbf{P} = [P(i, j)]$, $1 \leq i \leq a, 1 \leq j \leq a$. This probability matrix represents the percentage of time a given location (i, j) is occupied and can be computed by adding the corresponding steady state probabilities obtained from (3.33):

$$P(i, j) = \sum_{k,l} \pi_{(i,j;k;l)}, \quad (3.34)$$

where $(k, l) = \{(i-1, j), (i, j-1), (i+1, j), (i, j+1)\}$ for the non-boundary states. The (k, l) -pairs for boundary-states can be determined in a straightforward manner.

The transient coverage probability distribution, $\mathbf{P}^{(n)} = [P^{(n)}(i, j)]$, is computed similarly as:

$$P^{(n)}(i, j) = \sum_{k,l} \pi_{(i,j;k;l)}^{(n)} \quad (3.35)$$

Using the obtained $\mathbf{P}^{(n)}$, we can compute the probability that location (i, j) is covered by time step n as follows:

$$C^{(n)}(i, j) = 1 - \prod_{\nu=0}^n (1 - P^{(\nu)}(i, j)) \quad (3.36)$$

In the case of multiple drones, the state probabilities can easily be computed. Given the steady-state coverage distribution matrix of the drone k is \mathbf{P}_k (entries obtained using (3.34)) and assuming independent/decoupled mobility, the steady-state coverage distribution of an m -drone network can be obtained as:

$$P^{multi}(i, j) = 1 - \prod_{k=1}^m (1 - P_k(i, j)) \quad (3.37)$$

The transient behavior of the m -drone network can be computed similarly, by substituting the (i, j) -th entry of the transient coverage probability matrix ($\mathbf{P}_k^{(n)}$) (from (3.35)) into (3.37).

We now define some potential metrics of interest besides the coverage distribution of a mobility model in a grid: average coverage ($E\{C^{(n)}\}$) and full coverage probability ($\xi^{(n)}$) at time step n for a grid of size $a \times a$:

$$\begin{aligned} E\{C^{(n)}\} &= \frac{\sum_{i,j} C^{(n)}(i, j)}{a^2} \\ \xi^{(n)} &= \Pr(\mathbf{C}^{(n)} = \mathbf{1}_{a \times a}) = \prod_{i,j} C^{(n)}(i, j) \end{aligned} \quad (3.38)$$

where $\mathbf{1}_{a \times a}$ is an $a \times a$ matrix of ones.

These metrics carry some valuable information regarding the coverage performance, e.g., how well a given point is covered, how well the whole area is covered, or how much time would be necessary to cover the whole area.

In the next section, we provide the corresponding state transition probabilities for some representative independent and cooperative mobility models.

3.4.3 Transition Probabilities

Independent Mobility

In this section, we first shortly provide the state transition probabilities for the well-known random walk and random direction mobility models, where the transition probabilities are very intuitive. Note that for random walk the knowledge of the previous location is not necessary. Therefore, the states of the analytical tool $(i, j; k, l)$ can be further simplified to (i, j) , however, we omit this step for consistency with the other models. For random walk, we assume that at each time step, the drone can go to any one of the neighboring grid points with equal probability. Clearly, the number of neighboring points change depending on the location (see Fig. 3.33 for a representation of the different areas). On the other hand, for random direction model, the direction is changed only when the drone reaches the boundary of the grid. Therefore, the previous location, which is also equivalent to direction for the lattice, needs to be taken into account. For both of these schemes as well as the cooperative scheme proposed in the next section, at *the boundaries and corners* the next location is chosen randomly among the available neighboring points with equal probability. Table 3.6 shows the forward, backward, left, and right transition probabilities for random walk and direction models, respectively. The entries are organized as [transition probability, location, direction of movement].

C_1	B_2	C_2
B_1	M	B_3
C_4	B_4	C_3

Fig. 3.33. Location classification: corner (C_i), boundary (B_i), and middle (M)

Cooperative Mobility

In this section, we propose a method to approximate the coverage performance of a cooperative mobile network. In such a network, the nodes interact with each other (i.e., exchange information) whenever they meet. The amount or content of exchanged information is not within the scope of this section. The objective is to come up with an appropriate transition probability matrix that can be used by the proposed stochastic tool. Recall that the proposed Markov chain is for a single drone. For independent mobility, it can easily be extended to multiple drones. However, for cooperative mobility this Markov chain is

Table 3.6. Random walk (RW) and direction (RD)

	Corners	Boundaries	Middle RW	Middle RD
P_B	$1/2$ ($C_i \uparrow \rightarrow \downarrow \leftarrow$)	$1/3$ ($B_i \uparrow \rightarrow \downarrow \leftarrow$)	$1/4$	0
P_F	0 ($C_i \uparrow \rightarrow \downarrow \leftarrow$)	$1/3$ ($B_{i=1,3} \uparrow \downarrow$, $B_{i=2,4} \leftarrow \rightarrow$)	$1/4$	1
P_L	$1/2$ ($C_1 \leftarrow$, $C_2 \uparrow$, $C_3 \rightarrow$, $C_4 \downarrow$)	$1/3$ ($B_1 \leftarrow \downarrow$, $B_2 \uparrow \leftarrow$, $B_3 \rightarrow \uparrow$, $B_4 \downarrow \rightarrow$)	$1/4$	0
P_R	$1/2$ ($C_1 \uparrow$, $C_2 \rightarrow$, $C_3 \downarrow$, $C_4 \leftarrow$)	$1/3$ ($B_1 \leftarrow \uparrow$, $B_2 \uparrow \rightarrow$, $B_3 \rightarrow \downarrow$, $B_4 \downarrow \leftarrow$)	$1/4$	0

not sufficient to model the interactions. The states of a Markov-chain that exactly models all the interactions would grow exponentially with the number of drones. Therefore, in this section, we propose an approximate method to model the behavior of the drones in a way that would allow us to treat the cooperative mobility as independent mobility.

To “decouple” the actions of the drones from each other we define the following for an m -drone network:

$$P_X = \sum_{k=0}^{m-1} P_{X|k} \Pr(k+1 \text{ nodes meet}), \quad X \in \{B, F, L, R\} \quad (3.39)$$

where the backward, forward, left-turn and right-turn probabilities are given by the decision metric ($P_{X|k}$) of the cooperative mobility as well as the number of drones that meet. Clearly, probability of a meeting depends on the mobility model. However, for simplicity, in this work, we make the strong assumption that any node can be anywhere in the grid with equal probability. The implications of such an approximation will later be quantified by simulations. With this assumption, from the perspective of a drone at location (i, j) of a grid of size $(a \times a)$, probability that exactly k other nodes out of a total of m drones will also be at (i, j) is given by the binomial distribution:

$$\Pr(k+1 \text{ nodes meet}) = \binom{m-1}{k} \left(\frac{1}{a^2}\right)^k \left(1 - \frac{1}{a^2}\right)^{m-1-k} \quad (3.40)$$

The entries of the corresponding transition probability matrix can then be computed using (3.39) and (3.40), given the decision metric ($P_{X|k}$). If you have a cooperative rule quantified by decision metric $P_{X|k}$, these equations along with the analytical model from Section III can be used to quantify the coverage performance.

In the following, we provide an application of this method for simple cooperative mobility. It uses only the previous locations and number of the meeting drones in the decision criteria (e.g., as in [151]). The objective is to cover a given area as fast and as efficiently as possible. With such an objective, an

intuitive rule is that the drones move to a previously unoccupied location with a high probability. Clearly, since we consider only the previous direction, the final decision might be good only locally and might not contribute to global coverage. The mobility rules at a grid point (i, j) is summarized in Algorithm 3, where n_0 denote the number of unoccupied neighbors of (i, j) at the previous time step.

Algorithm 3 Cooperative Mobility Algorithm

1. If (i, j) is not in the boundaries or corners of the grid:
 - a) If $k = 0$, i.e., there is only one drone at (i, j) , the drone keeps going forward until it meets another drone or until it hits the boundary.
 - b) If $k \geq 1$, the drones determine the *unoccupied* neighbors, n_0 , of (i, j) at the previous time step and
 - i. If $n_0 \geq 2$, then the drones move to any one of the unoccupied grid points with probability $\frac{1}{n_0}$
 - ii. If $n_0 = 1$, then the drones move to the unoccupied grid point with probability p_0 and the other 3 occupied grid points with probability $\frac{1-p_0}{3}$. Clearly, if $p_0 = 1$, this rule is equivalent to above rule (i). A non-zero p_0 option is given to prevent all drones from moving into the same location.
 - iii. If $n_0 = 0$, then the drones move any one of the 4 neighboring grid points with probability $1/4$.
 2. If (i, j) is in one of the boundaries or corners of the grid, then the same rules as the independent random mobility models are applied regardless of the presence of a meeting.
-

Next, we derive the transition probabilities for the middle grid points. Observe that due to the symmetry of the decisions $P_{F|k} = P_{L|k} = P_{R|k}$, when $k \geq 1$. Therefore, if we compute $P_{B|k}$, all other probabilities would be determined as well. From the rules above, $P_{B|k}$ is non-zero only when $n_0 \leq 1$. To this end, we first compute the probability that $n_0 = 1$ and $n_0 = 0$ given $k + 1$ nodes meet.

There are 4^{k+1} different ways that $k + 1$ nodes can meet. Assume that each of these meetings happen with equal probability $\frac{1}{4^{k+1}}$. Using combinatorics for selection with repetitions, we can derive the probabilities that $n_0 = 1$ and $n_0 = 0$, respectively, when $k + 1$ nodes meet as follows:

$$\Pr(n_0 = 1) = \frac{\sum_{i=1}^{k-1} \sum_{j=1}^{k-i} \binom{k+1}{i} \binom{k+1-i}{j}}{4^k} \quad (3.41)$$

and

$$\Pr(n_0 = 0) = \frac{\sum_{i=1}^{k-2} \sum_{j=1}^{k-i-1} \sum_{l=1}^{k-i-j} \binom{k+1}{i} \binom{k+1-i}{j} \binom{k+1-i-j}{l}}{4^{k+1}}. \quad (3.42)$$

Table 3.7 presents these probabilities for different k values. As the number of nodes that meet, i.e., $k + 1$ increases, as expected, $\Pr(n_0 = 0)$ increases and correspondingly, $\Pr(n_0 = 1)$ starts decreasing after a certain point.

Table 3.7. $\Pr(n_0 = 0)$ and $\Pr(n_0 = 1)$

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$\Pr(n_0 = 0)$	0	0	0.09	0.23	0.38	0.51
$\Pr(n_0 = 1)$	0	0.375	0.56	0.59	0.53	0.44

Using (3.41) and (3.42), $P_{B|k}$ for $m > 3$ is given by:

$$\begin{aligned}
 P_{B|k} &= \sum_{i=0}^3 \Pr(n_0 = i) P_{B|ki} \\
 &= \begin{cases} 0, & k < 2 \\ \Pr(n_0 = 1) \frac{1-p_0}{3}, & k = 2 \\ \Pr(n_0 = 1) \frac{1-p_0}{3} + \Pr(n_0 = 0) \frac{1}{4}, & 2 < k \leq m-1 \end{cases} \quad (3.43)
 \end{aligned}$$

When $m < 3$, $P_{B|k} = 0$, and when $m = 3$, $P_{B|k} = \Pr(n_0 = 1) \frac{1-p_0}{3}$.

Finally, substituting (3.40) and (3.43) into (3.39) we can compute P_B . Similarly, P_F , P_R , and P_L can be computed substituting the following relations into (3.39):

$$P_{F|k} = \begin{cases} 1, & k = 0 \\ \frac{1-P_{B|k}}{3}, & 0 < k \leq m-1 \end{cases} \quad (3.44)$$

and

$$P_{R|k} = P_{L|k} = \begin{cases} 0, & k = 0 \\ \frac{1-P_{B|k}}{3}, & 0 < k \leq m-1. \end{cases} \quad (3.45)$$

Then, the transition probability matrix for each drone can be obtained using the derived P_X 's for the middle cells and the boundary/corner grid behavior described in the previous subsection. The overall coverage performance of an m -drone cooperative network can then be determined using (3.37).

3.4.4 Coverage Performance and its Discussion

In this section, we evaluate the validity of the proposed analytical method for several different scenarios by comparing the analysis with Monte Carlo simulations (where the coverage distributions are obtained by averaging over 10000 runs). For the cooperative mobility model, we use $p_0 = 0.25$. Different number of drones (m), grid sizes (axa), and time steps (n) are evaluated.

First, we evaluate the steady-state coverage distribution, which corresponds to the percentage of time a given point would be covered. Fig. 3.34 (a) and (b) show the average time coverage versus number of drones m and the grid dimension a . The steady-state coverage distribution matrix is computed using (3.33) and (3.34). Both the analytical and simulation results in the figure are then obtained by averaging overall points in the grid. Observe that the steady-state performance of all schemes are the same, shown by both simulation and analysis. While the limiting distributions of all the schemes are the same, the time required to reach this distribution varies between mobility models.

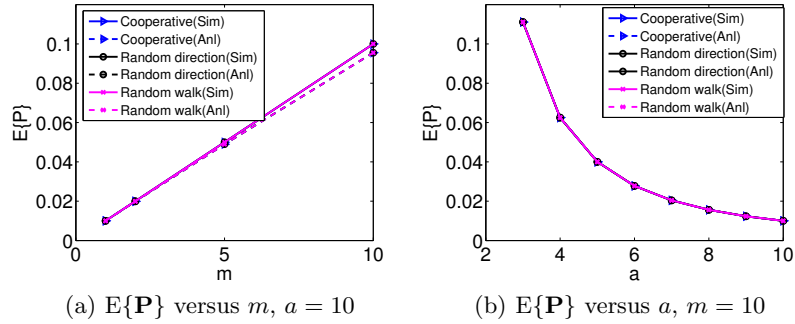


Fig. 3.34. Average steady-state (time) coverage.

Next, we look at the transient behavior of the mobility models under study. Fig. 3.35 shows a snapshot of the coverage at time step $n = 10$ for different number of drones and grid dimensions. As expected, the coverage increases with increasing number of drones and decreases with grid size. While for $a = 5$ coverage over 90% can be achieved with 10 drones, the achievable coverage drops below 40% when the grid size is increased to 10x10. Observe that the simulation and analysis results are in agreement in general. The highest deviation is observed as the number of drones increase. Recall that the average coverage is computed over all grid points, and hence, the deviations in the coverage of each grid point, however small they maybe, propagate and could become significantly large when added. Therefore, to check the similarity of the coverage probability distributions obtained from the analysis and the simulation, we use the following Euclidean distance metric:

$$MSE^{(n)} = E \left\{ (C_{ani}^{(n)}(i, j) - C_{sim}^{(n)}(i, j))^2 \right\}. \quad (3.46)$$

Fig. 3.36 presents the mean square error obtained using (3.46), when $n = 10$ and $a = 10$. Observe from these results that while the average coverage obtained from the analysis and simulation may deviate from each other, the

individual coverage of the grid points on average deviate around 0.18%. We are currently in the process of determining a distance metric that does not suffer from numerical approximation limitations to better identify the deviations.

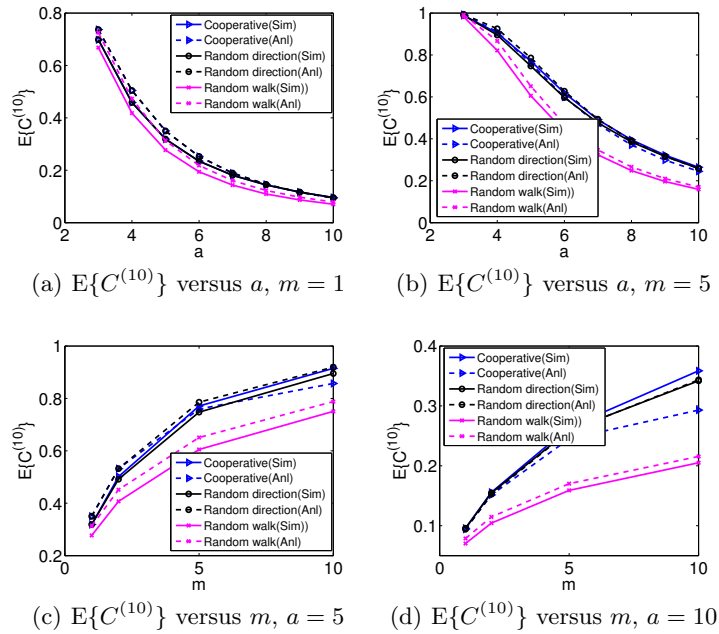


Fig. 3.35. Average spatial coverage when $n = 10$ steps.

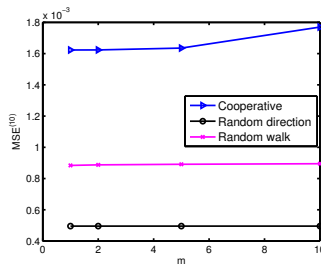


Fig. 3.36. MSE versus m , when $n = 10$ and $a = 10$.

Finally, we illustrate the progress of coverage with time. Fig. 3.37 shows the MSE, average and full spatial coverages (from (3.38)), when $a = 5$ and $m = \{1, 5\}$. The MSE reduces as the number of time steps increase and is less than 2% for $m = 1$ and less than 0.4% when $m = 5$. As a result, the average

and full spatial coverages from analysis and simulation also deviate from each other less, when $m = 5$. Comparing average and full coverages, we observe that while the likelihood that each point is covered on average can be above 99% around $n = 200$ (when $m = 1$), full coverage requires significantly more time. Therefore, a threshold-based coverage metric can be more suitable than average or full coverage for some applications. Nevertheless, the analytical tool can provide some insight into how much time would be required to achieve a certain coverage level and allows for testing different performance metrics of interest.

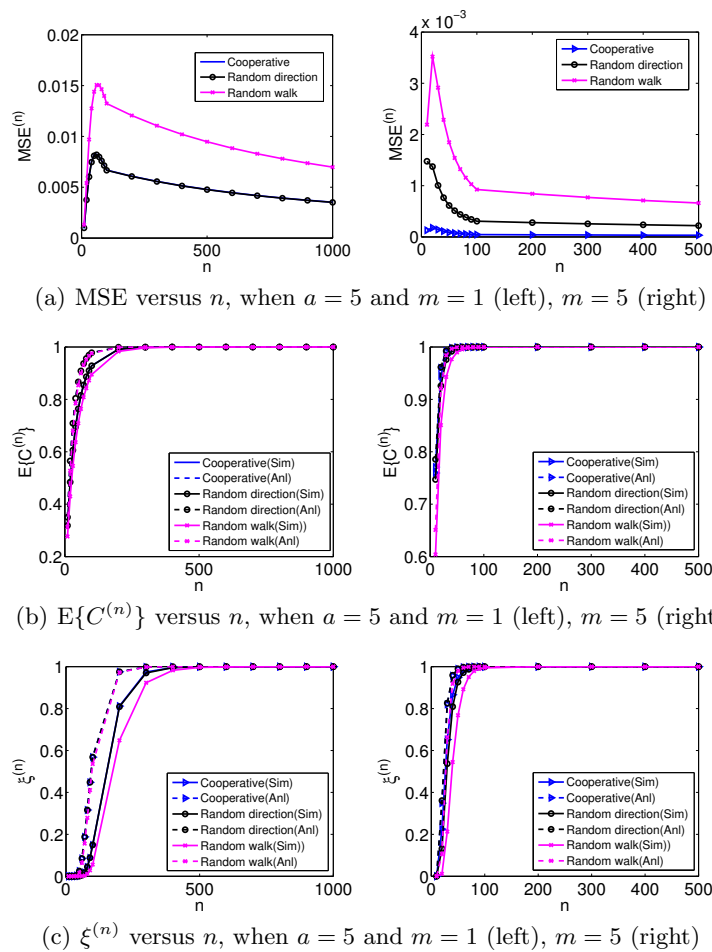


Fig. 3.37. Transient behavior comparison

3.4.5 Conclusions

In this work, we proposed an analytical model to evaluate the coverage performance of a networked UAV system. We showed the validity and the limitations of the analytical tool by comparing with simulations for several scenarios. The performance metrics of interest focused on in this section were coverage distribution, average and full coverages. We observed that while the coverage distributions can be estimated well with the analytical model, the average and full coverages can deviate from the simulations for certain system parameters due to error propagation.

3.5 Conclusions

In this chapter we introduced some schemes of placements for nodes, that are optimal for energy consumption. In order to further improve the lifetime of the network we proposed some distributed and efficient algorithms for movement coordination among nodes. Afterwards, some features of these algorithms have been added to a routing protocol that is able to choose opportunely the path for data forward taking into account the residual energies and the distances to travel towards the final optimal positions. All algorithms and protocols proposed show the impact of mobility on some performance parameters of the networks. We proposed an analytical tool to valuate the coverage obtained using different mobility patterns, while focusing on the coverage in UAV networks.

Bio-Inspired mechanism for Self-Organization

4.1 Introduction

All methodologies and algorithms that we have used until now are distributed but they are not reactive in regard to unforeseen situations. Furthermore, we have assumed that all the capabilities of nodes are embedded, nothing can be learned from the environment.

In this chapter we want to overcome these limitations using mechanisms inspired by biology, such as swarm intelligence and evolving neural networks in order to move towards Self-Organization in Wireless Sensor, Robot, and UAV Networks.

The social insect metaphor for solving problems has become a hot topic in the last years. This approach emphasizes distributedness, direct or indirect interactions among simple agents, flexibility and robustness. In practice, Swarm Intelligence is a methodology to solve many problems and it is inspired to swarms of animals such as ants, bees, birds, etc. The single animal typically has limited capabilities, but a group of individuals is able to solve complex problems such as the nest building, or the path search from a food source to the nest, etc. Many research activities conducted to design metaheuristics based on the behaviors of natural swarms.

Furthermore we want to introduce a process of learning that allows nodes to infer new behaviours in order to meet the dynamics of the surrounding environment. In this chapter we will use reinforcement learning algorithms within an evolving neural network, based on a genetic algorithm used to determine the weights of neurons connections.

The remainder of the chapter is organized as follows: in section 4.2 different techniques for coverage issues based on Virtual Forces and Particle Swarm Optimization are investigated. In sections 4.3 and 4.4 two different evolving neural networks are proposed for optimal placement and efficient mobility patterns, respectively, both of them focus on achieving full coverage of an area of interest.

4.2 Distributed Algorithm to Improve Coverage for Mobile Swarms of Sensors

This section focuses on Mobile Wireless Sensor Network. Mobility of the devices is achieved by the design of algorithms based on Particle Swarm Optimization (PSO) and Virtual Forces. Mobility is exploited in order to improve both the power consumption and the coverage of specific Zone of Interest (ZoI) that can change dynamically. We will show how the proposed algorithms are reactive, i.e. they are able to capture in an effective fashion the events in the sensor field even if the position of the events changes during the simulation. We modify the classical PSO approach in order to be able to design totally distributed algorithms, which need only local information from neighbors to update the velocity of the devices (PSO-S). This new distributed version of the PSO is combined with a distributed version of the Virtual Forces Algorithm (VFA). Furthermore, we also design and propose a distributed implementation of the VFA that we call VFA-D. In order to show the effectiveness of the the proposed techniques, we perform many simulations to compare the PSO-S and the VFA-D schemes with a centralized version of the VFA. Simulation results show the good performance in terms of coverage and energy consumption as well as the high reactivity of both PSO-S and VFA-D when the ZoI changes.

4.2.1 Introduction

Swarm Intelligence is a new discipline inspired by the behavior of social bugs or animals such as ants, bees, birds, fish, etc. [155] The main characteristic of this kind of animals is that individually they have limited capacity, but in a group they are able to perform very complex tasks, including nest building, navigation, foraging, food storage, tending the young, garbage collection, etc. In the last years, animal systems have been the inspiration source for several heuristics used to solve hard optimization problems in various disciplines. One of the most used heuristic is the Particle Swarm Optimization (PSO) [154]. Specifically, the PSO can be considered as a stochastic optimization technique based on flocks of birds and schools of fish. In the PSO, a set of software agents called *particles* are placed in a search space of an optimization problem and move in this space according to certain rules and global information to find the best solution for the problem. We are convinced that this approach can be beneficial for several Mobile Wireless Sensor Network (MWSN) problems. Specifically, we focus on the problem of dispatching mobile sensors in a field of interest to monitor dynamic events. No knowledge about either the position or the duration of the events is given *a priori*. Thus, mobile sensors have to discover the events, monitor them and move towards a new Zone of Interest (ZoI) when the previous monitored event is over. An efficient, distributed and localized solution of this problem would be immediately exploitable by several applications domains, such as civil, environmental, military, etc.

This section proposes a modified version of the PSO, where particles (mobile sensors, nodes or devices in the following) update their velocity by using only local information coming from their neighbors. In practice, the update of the velocity is performed by the means of a consensus algorithm, a well-known technique in the field of multi-agent systems. The concept of the “neighbor” is intended in the classical telecommunications sense, that is two nodes are neighbors if they are placed within the communication range of each other. Our modified version of the PSO is also integrated with a distributed version of the Virtual Force Algorithm (VFA). The Virtual Force technique is able to drive nodes in a way that there is no overlap in the position of sensors, by using attractive and repulsive forces based on the distance between the devices. We also proposed, test and evaluate the performance of the distributed version of the VFA by itself. The techniques we propose are able to reach high levels of coverage and show a satisfying reactivity when the ZoI changes. This last output parameter is measured in terms of the capability for the sensors to “follow” a sequence of events happening in different ZoI.

The main contributions of this work can be summarized as follows:

- We propose a modified distributed version of the classical PSO (modified PSO);
- We propose a modified distributed version of a classical Virtual Force approach (VFA-D);
- We propose a “combined” version (PSO-S) of the modified PSO and the VFA-D.

In order to show the effectiveness of our techniques we perform a series of simulations and we compare our algorithms with the classical VFA that needs a centralized unit to collect from and distribute to the nodes global information about their positions [165].

4.2.2 Related Work

Wireless Sensors Networks have been successfully considered for many applications such as monitoring and surveillance [156]. In [12] authors introduce the concept of Virtual Forces in order to obtain an improved coverage of a specific geographic area. Initially, nodes are randomly deployed and then they are subject to attractive and repulsive forces from other nodes that makes them move toward a new position, where the area coverage is maximized. Even though VFA is a very efficient algorithm for coverage, some improvements have been proposed in recent years, in order to enhance connectivity and fault tolerance while saving energy [158], and provide self-repairing and anti-splitting abilities[161]. A common assumption in these works is that all nodes exert forces on every other node, thus requiring global information on nodes positions. In our work, we remove this assumption and use only neighborhood information. Authors of [164] and [157] use evolutionary and

learning mechanisms to let nodes find their best placement, which does not make the schemes suited when the ZoI change dynamically. The PSO technique has been considered with different objectives and in different scenarios [159]. In telecommunications, many applications focus on clustering in ad-hoc networks in order to minimize the energy consumption [153]. by using Voronoi and PSO jointly. Even this scheme does not consider the possibility for the nodes to move in reaction to a topological change of the events placement.

4.2.3 Proposed Algorithms

In this section we give the details of the techniques we designed. Before considering the specific techniques we recall some elements of both the classical VFA and the PSO, useful to understand the difference with our proposals.

Centralized Virtual Force Algorithm

The Virtual Forces Algorithm is based on the concept of the virtual forces field and the main objective is the maximization of the coverage in a Wireless Sensors Network (WSN) [165]. In the same way electro-magnetic particles attract or repel each other based on the value of the potential field, sensor nodes attract or repel each other based on their mutual distance.

The model presented in [165] is the following:

$$\mathbf{F}_i = \sum_{j=1, j \neq i}^k \mathbf{F}_{i,j} + \sum_{m=1}^M \mathbf{F}_{iR_m} + \sum_{n=1}^N \mathbf{F}_{iA_n} \quad (4.1)$$

where the total force exerted on node i is given as the sum of three forces:

1. $\mathbf{F}_{i,j}$ is the total force (attractive and repulsive) that all the k nodes present in the field exert on node i ;
2. \mathbf{F}_{iR} is the total repulsive force that all the M obstacles exert on node i ;
3. \mathbf{F}_{iA} is the total attractive force that all the N areas to be monitored exert on node i .

The first force can be expressed by the following formula:

$$\mathbf{F}_{i,j} = \begin{cases} (w_A(d_{ij} - d_{th}), \alpha_{ij}) & \text{if } d_{ij} > d_{th} \\ 0 & \text{if } d_{ij} = d_{th} \\ (w_R(\frac{1}{d_{ij}}), \alpha_{ij} + \pi) & \text{if otherwise} \end{cases} \quad (4.2)$$

where d_{ij} and α_{ij} are respectively the Euclidean distance and the angle between nodes i and j , d_{th} is the threshold distance for nodes to attract or

repel each other, w_A and w_R are the weights of the attractive and repulsive forces, respectively. The novel position is calculated in [160] from, F_{xy} the magnitude of \mathbf{F}_i and its x and y components, F_x and F_y , respectively, as follows:

$$x_{new} = x_{old} + \left(\frac{F_x}{F_{xy}}\right) \cdot MaxStep \cdot e^{\frac{-1}{F_{xy}}} \quad (4.3)$$

$$y_{new} = y_{old} + \left(\frac{F_y}{F_{xy}}\right) \cdot MaxStep \cdot e^{\left(\frac{-1}{F_{xy}}\right)} \quad (4.4)$$

where $MaxStep$ is the predefined single maximum moving distance.

In the classical version of this algorithm, a central entity is requested to collect all the information from the nodes in order to compute the total force exerted on each of them, for this reason we will refer to this technique as VFA-C (Virtual Force Algorithm - Centralized). Besides the complexity and the problems caused by a single point of failure, introduced with a central coordinator, we will show that the VFA-C fails in the case that ZoI are on opposite sides of the field.

The Distributed Virtual Force Algorithm: VFA-D

In this work we modified and implemented by simulation a new version of the VFA that avoids the usage of a central entity to calculate the forces because it does not require global information for the computation of the forces applied on the nodes. In fact, VFA-D introduces a maximum distance in order for the nodes to retrieve information. This maximum distance, which we call C in the following, is related to the sensing range of the nodes. When another node, an obstacle or a ZoI are farther than C from the current node, then their effects on the node are considered negligible and will be neglected.

The Particle Swarm Optimization Algorithm (PSO)

The Particle Swarm Optimization is an extremely versatile technique of swarm intelligence based on the position of the particles [154]. The particles of the PSO are localized inside a searching space and evaluate an objective function depending on their own position. These particles can also move around the searching space and combine their own knowledge with those received from one or more neighbors. By assuming that particles move in a 2D searching space, the velocity of the units will be computed iteration by iteration as follow:

$$\mathbf{v}_i(t+1) = \omega \cdot \mathbf{v}_i(t) + \phi_p \cdot \mathbf{r}_p \circ (\mathbf{p}_i - \mathbf{x}_i(t)) + \phi_g \cdot \mathbf{r}_g \circ (\mathbf{p}_g - \mathbf{x}_i(t)) \quad (4.5)$$

where $x_i(t)$, $v_i(t)$, p_i , p_g , r_p and r_g are R^2 vectors. Specifically, $x_i(t)$ and $v_i(t)$ are the current position and the velocity of the particle i ; p_i is the best personal position of the particle, p_g is the best position of the swarm; r_p and r_g are two random vectors in the domain $U(0, 1)$; w , ϕ_p and ϕ_g are selected parameters to control the efficiency of the PSO technique and \circ is the Hadamard multiplicative operator. The three components are also referred as *inertia*, *cognitive component* and *social component*, respectively. The new updated position of the particle at the next step is:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (4.6)$$

where the new position is given incrementally from the previous position when the new velocity has been applied in the time instant under observation.

The Modified Particle Swarm Optimization: Modified-PSO

We designed and implemented our modified PSO by using the concept of *consensus*. The term ‘‘consensus’’ in multi-agent systems indicates the process of reaching an agreement on a certain quantity of interest that depends on the state of all agents. A consensus algorithm states the information exchange between an agent and all of its neighbors in the network [162]. Usually, in the PSO scheme, the social component is the best position achieved globally by the swarm in the research space. For our matters, it is not useful to consider a global best position, because it implies a centralized scheme of control or, at least, the capacity of the nodes to communicate with every other node in the sensor field. In order to take into account the limited communication capabilities of sensors, we stated that the social term involves the position that enjoys the maximum consensus within each node’s neighborhood, where a neighborhood is composed only of the sensors within its transmitting/receiving range. Thus, we assume that at each iteration of the modified PSO algorithm, the sensors exchange information and determine the maximum consensus in their neighborhood. The velocity update equation is modified as follows:

$$\mathbf{v}_i(t+1) = \omega \cdot \mathbf{v}_i(t) + \phi_p \cdot \mathbf{r}_p \circ (\mathbf{p}_i - \mathbf{x}_i(t)) + \phi_g \cdot \mathbf{r}_g \circ (\mathbf{l}_i) \quad (4.7)$$

where:

$$\mathbf{l}_i = \frac{\mathbf{x}_k - \mathbf{x}_i}{\|\mathbf{x}_k - \mathbf{x}_i\|} \cdot \frac{\|\mathbf{x}_k - \mathbf{x}_i\|}{d_{rep}} \quad (4.8)$$

In 4.8, x_k is the position of the nodes in the set of neighbors of i that obtained the best value of the objective function in the previous iteration and d_{rep} is a repulsive coefficient that avoids the overlap of nodes. The inertial weight w varies between w_{max} and w_{min} .

The Serial Particle Swarm Optimization: PSO-S

This algorithm is designed by considering separately the modified PSO and the VFA-D. Specifically, we first apply the modified-PSO and when a sub-optimum solution is achieved, we apply the VFA to optimize the final coverage solution. Since the resulting algorithm applies the two presented schemes in a serialized way, we named it PSO-S (PSO - Serialized). Of course, in this case we need to specify the exact times of stop (for the Modified-PSO) and start (for the VFA-D). We formulate three convergence conditions, which will be used in the simulation:

1. nodes position does not change significantly in last iteration (travelled distance smaller than d_c),
2. nodes coverage does not change significantly in a certain number of iterations (coverage improvement smaller than c_{th}),
3. nodes have already run the algorithm for a certain number of iterations (number of iterations larger than it_{max}).

When any of the previous condition is verified the Modified-PSO stops running and the VFA-D enters into action. Specifically, the third condition is useful when nodes keep moving without finding an optimal solution.

4.2.4 The Simulation Environment

In order to evaluate the effectiveness of the algorithms proposed, we considered a square sensor field of $100m \times 100m$, where a certain number of events occur simultaneously, and a fixed number of mobile sensors are randomly placed. We assume that the ZoIs change dynamically during the simulation time, in order to simulate a sequence of events that appear and disappear in the field.

Our first objective is to achieve a high level of coverage for the ZoI where events occur, but we also evaluate the energy consumption for the movement. The energy model for the movement used for this work is taken from [163], it takes into account the traveled distance by nodes by a constant k ranging from 0.1 and 1 J/m . We set this constant equal to 0.1 J/m . The most interesting parameters used in the simulation are summarized in Table 4.1.

In order to evaluate the effectiveness of the techniques we proposed in this work we figured out many particular scenarios with different and specific events distributions. In Figure 4.1, we show the simulated scenarios.

The white zones represent the ZoIs, which are the areas of the field where events happen and have to be monitored, the black zones are areas where no events occur.

The choice of these specific scenarios is related to the capability of the technique we designed to adequate in a dynamic fashion to many different situations and we will show how our algorithms are able to capture the events in a distributed fashion.

Table 4.1. Evaluation Parameters

Field Area ($L \times L$)	100 m x 100 m
Number of Mobile Sensors (N)	80
Sensing Radius (R_s)	7 m
Communication Radius (R_c)	$2r_s$
Convergence Distance (d_c)	0.5 m
Maximum Number of Iterations (it_{max})	500
Threshold Coverage (c_{th})	0.5 – 0.9
Inertia Weights ($w_{min} - w_{max}$)	0.1 – 0.7
Attractive Force (w_A)	0.01
Repulsive Force (w_R)	1000
Coverage Threshold (c_{th})	0.5 – 0.9
Threshold Distance for forces among nodes (d_{th})	$2r_s$
Threshold Distance of virtual force vanishing (C)	$4r_s$
Repulsion Coefficient for PSO (d_{rep})	$2r_s$
Confidence Interval	95%
Number of Runs	100

Simulation Results

In this section we show the behavior of the different techniques considered in this work: the Virtual Force (VFA-C), the distributed version of the Virtual

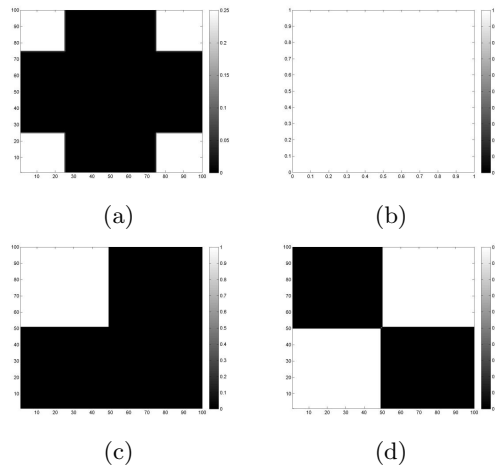


Fig. 4.1. Simulated scenarios. (a) Scenario 1: events are concentrated in the corners, (b) Scenario 2: events are uniformly distributed, (c) Scenario 3: events are concentrated in one square, (d) Scenario 4: events are concentrated in two squares.

Force algorithm (VFA-D), the combined version of the Modified-PSO and the VFA-D (PSO-S). In order to evaluate the effectiveness of these techniques, we consider three output parameters: 1) the coverage, as the fraction of ZoI covered by sensors, 2) the energy consumed by nodes movement, which represent the cost of the algorithm and 3) the number of iterations, which helps understanding the reactivity of the algorithm. As we assume a probabilistic model for Virtual Forces [165], a generic point of the field can be considered *covered* only when its coverage is larger than a certain coverage threshold, the evaluation of these parameters is made by varying the coverage threshold c_{th} .

It is worth to recall that the algorithms terminate for any of the three different conditions enumerated in Section 4.2.3. In Figures 4.2 and 4.3 we show the results for Scenario 1, in terms of coverage and energy consumption achieved by VFA-C, VFA-D and PSO-S. We can notice that in this particular scenario the VFA-C fails completely to cover the ZoIs. This behavior is related to the centralized characteristic of the algorithm. In fact, each node exerts a force on every other node in the field and the “extreme” situation considered does not allow the nodes to find a correct position.

In Figure 4.3 we show the energy spent by each technique. Not only the VFA-C fails to accomplish the coverage task, also it spends more than the other two approaches. This is due to the fact that nodes move a lot in order to find a good position to capture the events.

This is confirmed from Figure 4.4, where we can notice that the algorithm VFA-C converges only when it reaches the maximum number of iterations fixed to it_{max} .

There are not specific reasons regarding the ZoIs we considered for changing the scenario. Simply, we considered many different changes of scenario and we present in this work the most meaningful to evaluate the behaviors and the characteristic of the proposed techniques.

In Figure 4.5 we show the coverage achieved by the three algorithms, when the position of the events changes from Scenario 4 to Scenario 3.

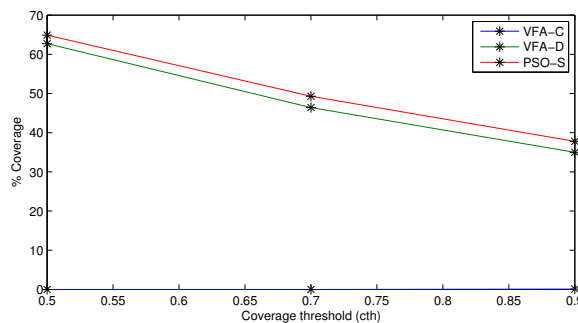


Fig. 4.2. Coverage achieved for Scenario 1.

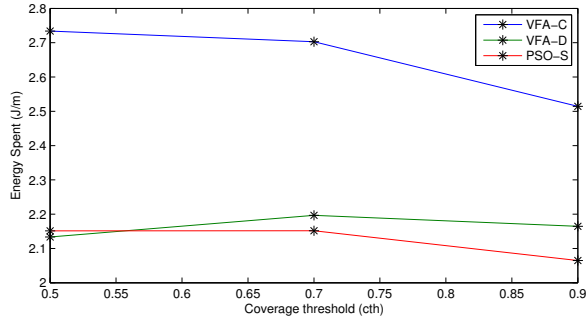


Fig. 4.3. Energy spent for movement in Scenario 1.

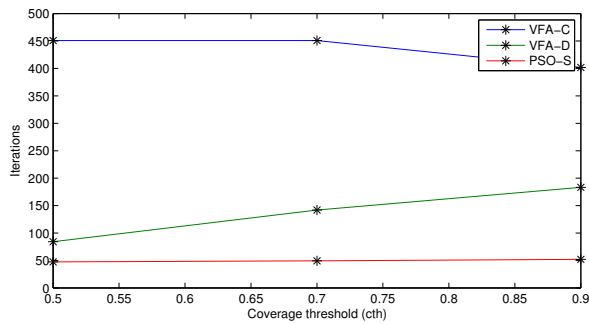


Fig. 4.4. Number of iterations to reach the convergence in Scenario 1.

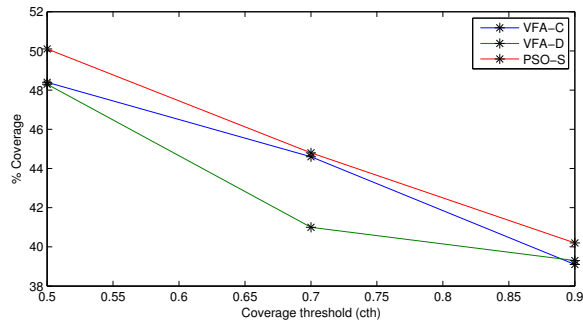


Fig. 4.5. Coverage achieved when the scenario changes from Scenario 4 to Scenario 3.

We can observe that for this specific change of the ZoIs, the VFA-C is able to reach good results in terms of coverage. The evaluation of the energy in this case shows that the three algorithms make nodes travel similar distances,

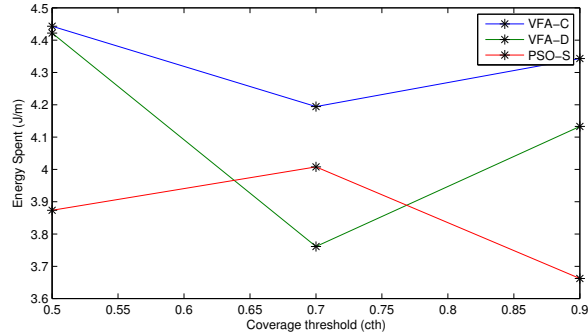


Fig. 4.6. Energy spent when the scenario changes from Scenario 4 to Scenario 3.

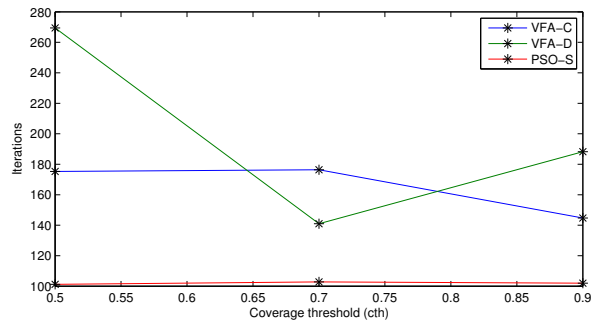


Fig. 4.7. Number of iterations when the scenario changes from Scenario 4 to Scenario 3.

as observed in Figure 4.6. In practice, the cost of the algorithms is similar for this change of scenario.

In Figure 4.7 we report the number of iterations each technique needs in order to converge towards the best solution. We can observe that the average number of iterations for convergence is similar when the c_{th} changes for VFA-C and PSO-S but changes significantly when the VFA-D is considered.

In Figure 4.8 we show the coverage for VFA-C, VFA-D and PSO-S, when the ZoI change from Scenario 2 to Scenario 1. Once again the VFA-C fails to accomplish the coverage task while the other two techniques are able to cover sufficiently the ZoIs.

In Figure 4.9 we consider the energy spent when the ZoIs change from Scenario 2 to Scenario 1. VFA-D and PSO-S have similar behaviour in this case, while VFA-C spends more energy without covering the ZoIs.

On the other hand, in Figure 4.10 we can observe that VFA-C terminates when it reaches the maximum number of iterations. The fastest algorithm is the PSO-S, which, in different situations, employs the same number of itera-

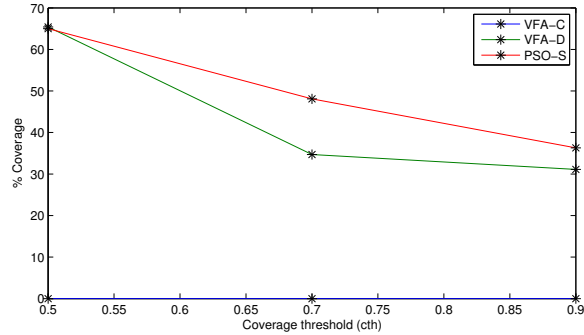


Fig. 4.8. Coverage achieved when the scenario changes from Scenario 2 to Scenario 1.

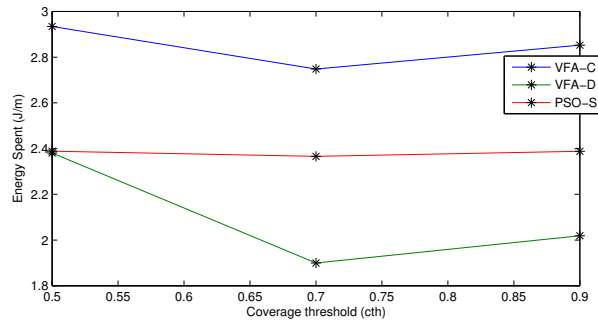


Fig. 4.9. Energy spent when the scenario changes from Scenario 2 to Scenario 1.

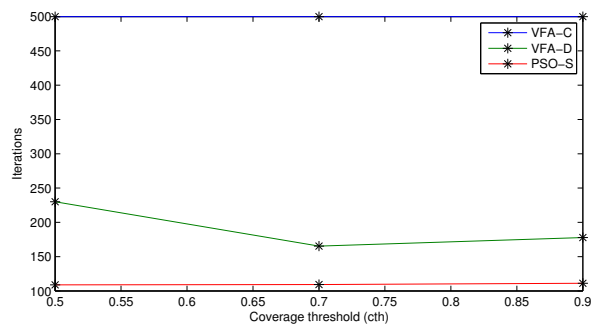


Fig. 4.10. N of iterations when the scenario changes from Scenario 2 to Scenario 1.

tions to converge. This suggests that the PSO-S is the most stable algorithm among those tested.

In conclusion, we saw how the VFA-C is not robust for all the scenarios and above all when the ZoIs changes in a dynamic fashion. On the other hand, our techniques show an inherent capability to make nodes adapt their position to the variations of ZoIs position.

4.2.5 Conclusion

In this section we considered particular distributions of events in a sensors field. We assumed mobile sensors able to communicate to each other. The specific task of the sensors is the coverage of portions of a square area where events occur. Specifically, we considered a known coverage technique based on potential field, the centralized Virtual Force Approach (VFA-C). Furthermore, we introduced a distributed version of the VFA-C, the VFA-D and a technique that combines a modified and distributed version of the PSO and the VFA-D, the PSO-S. Through extensive simulations we show the effectiveness of the new techniques to cover the Zones of Interest (ZoI), that is, the zones where the events occur. Even when these zone are at the extreme of the sensor field and the distributions of the events change in a dynamic fashion, our technique showed remarkable performance. In practice, we show the reactivity as an inherent property of these techniques. In future work, we would try to determine an automatic association between the choice of the weights of both the PSO and the VFA-D.

4.3 Nodes self-deployment for coverage maximization using an evolving neural network

Wireless Sensor and Robot Networks (WSRN) are composed by a high number of nodes that need to cooperate to accomplish several tasks (i.e. detect the presence of a malicious intruder in a region of interest, monitor the occurrence of a specific event, etc.). They have been applied in military applications and health care, as well as environmental monitoring and smart agriculture [62]. One of the fundamental issue in such a network is coverage. It is used to determine how well an interest area is monitored and a service can be provided. Nodes deployment is a critical issue for coverage, because it affects costs and detection capability of a WSRN. In literature, existing nodes deployment algorithms can be classified into the following categories: 1) Stationary sensors [63], [64] 2) Mobile Sensors [65] [13] and 3) Mobile Robot [66] [67]. Usually stationary sensors are random deployed in the area of interest. Random deployment is very simple, but the number of nodes to deploy has to be much larger than what is actually required for the full coverage. Furthermore, a random deployment of static nodes could be inefficient since some areas could be densely deployed in respect of others and some coverage holes or network partitions can occur. In some work [65] [13], a large number of stationary

nodes and a few mobile sensors are considered. Unfortunately, when the initial deployment of static nodes is random, this kind of approach is not useful to overcome the hole problems. In order to face these issues, it is possible to use only mobile nodes, but in this case it is necessary to implement an efficient and effective technique to allow nodes to move towards a certain position, in order to ensure the best coverage with the smallest amount of movement. In fact, energy consumption and probability to lose connectivity increase when nodes movement increases. These schemes are also known as movement-assisted techniques. An alternative is to use robots for deploying static nodes in a given region. Robot deployment can achieve full coverage with fewer nodes and can guarantee connectivity too. However, the presence of obstacles can challenge robot deployment and have a great impact on the deployment efficiency. To reduce the impact of obstacles in [67] authors propose four traveling orders for the robot movement. Unfortunately, this approach cannot guarantee full coverage and may introduce several sensing redundancies when the robot encounters obstacles. The approaches for coverage can be categorized into three groups: 1) force based [74], 2) grid-based [68] and 3) computational geometry based [69]. All the three categories can be considered as a sub-class of the movement-assisted category. With the focus on the self-organizing capabilities of nodes in WSRN, we propose a movement-assisted technique for nodes self-deployment. Specifically, we propose to use of a neural network as a controller for nodes mobility and a genetic algorithm for the training of the neural network through unsupervised learning. This kind of scheme is extremely adaptive, since it can be easily modified in order to consider different objective and/or QoS parameters. In fact, it is sufficient to consider a different input for the neural network to aim to a different objective. In this work we will show how the scheme works by considering the neural network as a controller for nodes mobility, when the objective is to maximize the coverage area and minimize the number of time steps to achieve the objective. To the best of our knowledge for deployment of wireless nodes able to self-organize, this is the first work based on neural networks and genetic algorithms. This technique has the clear advantage to make nodes able to learn from the environment and adapt their behavior. Based only on local information, nodes ensure a high coverage even when obstacles of irregular shapes are present in the area of interest. Another evident advantage of our algorithm is its simplicity: by the simple introduction of a new input, a different QoS parameter or objective can be taken into account. In respect of other approaches, our algorithm can be considered as a synthesis. In fact, our technique is grid based since it focuses on an interest area divided into cells. It can also be considered as a kind of virtual-force scheme based, because nodes are attracted or rejected from certain positions depending on the presence of other nodes. Finally, our technique belongs to the computational geometry based approach because the only local information needed at the nodes is the relative distance with other nodes and obstacles, and this leads to spatial

distributions of the nodes in the area of interest that follow geometric rules (e.g., uniformly spaced position).

The contributions of this work are the following:

- the approach makes behavior emerge from the interaction instead of being pre-programmed in the nodes;
- the methodology used in this work exploits both the flexibility of neural networks and the capability to learn of genetic algorithms, which makes it suitable to implement all the self-* properties;
- the proposed solution is simple, effective, distributed and feasible, even in presence of obstacles. It does not require any specific hardware/software that is not already included in mass products and needs only local information to work.

This work is presented in [3].

4.3.1 State of art

The problem of coverage in WSN and WSRN is closely related to deployment, because a good deployment can improve all the functionalities of the network. Indeed, Meguerdichian in [70] argues that coverage is the primary metric that provides indication about quality of service.

An important categorization regarding deployment for wireless networks as summarized in [71], classify algorithms as belonging to random deployment, incremental deployment and movement-assisted deployment. Random deployment is the fastest and most practical way to deploy a network, even if it does not ensure a uniform distribution. For this reason it is often used, as in our case, only in the initial phase of the movement-assisted deployment. Incremental deployment is a centralized approach, which places nodes one at time. The computation of optimal location for each node is based on information gathered by nodes already deployed. Hence, computation and time costs explodes when nodes number increases. In the last years, the most used approach to deploy a network is the movement-assisted, because it can achieve a uniform coverage with reasonable time and costs.

In [72] and [73], the approaches for coverage are categorized in three groups: force based [74] [165], grid-based [68], [75], [76] and computational geometry based [77], [78] and [69]. All these approaches can be considered as a sub-group of the movement-assisted approach.

In particular for the first group, in [165] authors propose a virtual force algorithm (VFA) to improve the coverage after an initial random placement over a region with obstacles. Even if their method works in presence of obstacles, it seems massively dependent on the mobility and energy of the sensors. Also authors of [79] consider obstacles. They develop an efficient technique to place sensors by means of a robot. Algorithm in [69] first deploys sensors along the boundaries and the region based on the sensing radius of the sensors. Successively, they apply delaunay triangulation and add some sensors in

order to reach full coverage. Both in [79] and [69], the methods developed seem suitable for simple regular regions and obstacles, like areas consisting of long straight lines. When obstacles and regions become sufficiently and arbitrarily irregular, these methods become inefficient. To consider obstacle in [12], authors introduce the concept of virtual force. Using this kind of Coulomb force proportional to the distance with other nodes or obstacles each node is able through attraction and repulsion to maximize the covered area. This is a cluster-based approach, so it shows all the drawbacks related to a centralized scheme. In [76] the whole region is divided into single-row and multi-row regions, in order to guarantee both coverage and connectivity. Specifically, holes are covered through sensors deployed along the boundaries, and connectivity is preserved by maintaining a constant distance computed from sensing and communication radii. Also this method seems suitable only for regular regions and obstacles.

Concerning the Grid quorum based movement [65], typically the sensor field is partitioned into many cells and the number of sensors in a cell is the load of the cell. Coverage and connectivity depend on the size of the cell. This deployment strategy cannot guarantee connectivity and cannot provide Points of Interest (PoI) coverage.

Regarding the computational geometry based approaches, in [13] authors propose an algorithm based on Voronoi regions. Each node, after the computation of its own Voronoi region, compares it with its sensing range in order to determine the presence of holes. In this case, the node moves to an improved location and recalculates the new Voronoi region until convergence is reached. This approach provides a uniform deployment in a distributed way, but is not suitable to use in scenarios with obstacles. Another distributed algorithm, known as pull & push [80], produces a hexagonal tiling, by attracting the nodes in low density regions and repulsing nodes from high density regions. This algorithm does not require manual tuning of variables related to the particular working scenario, even if it works only in absence of obstacles. Coulomb's law is also the base of [81], in this work authors propose a distributed algorithm inspired by equilibrium of molecules to obtain a uniform coverage. Each node calculates its lowest energy point and the distance from other nodes in a distributed manner, all computation requires exact location information and does not take into account the presence of obstacles.

The technique we propose in this work can be considered as a movement-assisted approach, in which the region of interest is sub-divided into specific cells (grid-quorum approach) and by the usage only of node-node and node-obstacle distance (computational geometry approach) attracts or repulses other nodes (virtual force approach) towards the best position for improving the coverage. A desirable property of our technique is its evolutionary nature, namely, its behavior varies in a "natural" fashion depending on the evolution process of the network. In previous schemes, such as Voronoi and VF schemes, the behavior is imposed in a pre-programmed fashion. Instead, our approach allows self-organization of nodes and it does not need any prior knowledge of the

environment, even in presence of obstacles. The novelty is that the deployment is not based on some geometric or physic rule pre-implemented in the nodes but the way to behave is learned directly from the environment during the training phase. The rules of movement learned are very general and not strictly related to any particular scenario.

4.3.2 A Genetic Algorithm for Nodes Self-Deployment

In order to solve the coverage problem through self-deployment of nodes in a wireless sensor and robot network, we propose to use neural networks and genetic algorithms in a combined approach. The neural network is used to model the behavior of the single node, whereas the objective of the genetic algorithm is to select the most performing population of nodes as a whole. In Section 4.3.3, we will show the effectiveness of the combined algorithm. Instead, in this Section we will illustrate the simplicity in the usage of the two main elements of the combined approach and we will introduce background and assumptions used in this work. The flowchart in Figure 4.11 describes the most relevant steps of the nodes self-deployment algorithm. The first three steps illustrate the initialization of simulation setup, neural network and genetic algorithm, while the following four contain the core algorithm.

In the following subsections we will refer to the flowchart as a general scheme of the proposed algorithm and we will provide details on each step.

Background and assumptions

The model proposed in this work is discrete in both time and space. We decided to use discretization for sake of simplicity, even though the method could be easily extended to a continuous model, where we would expect even better results at the cost of some complication in the computation required from the nodes. Hence, the field to cover is a square grid constituted of square cells, where also physical obstacles can be present. We consider one cell and one time step as discrete units of space and time, respectively. Also the sensing radius r of the nodes becomes a positive integer, and it expresses the number of cells that nodes are able to cover in each of the four main direction (north, south, east and west). Therefore, the coverage area of a node is a square area composed of $(2r + 1)^2$ cells. In Figure 4.12, the area covered by a node is coloured in light green. Cells coloured with a higher intensity of green have a higher degree of coverage (also referred to as k -coverage).

In the described scenario, we use the following assumptions regarding the behavior of nodes:

- In one time step nodes can move of one cell in one of the four main direction (north, south, east and west).
- Nodes are able to estimate the distance (in number of horizontal and vertical cells) between themselves and other nodes, and between themselves and obstacles.

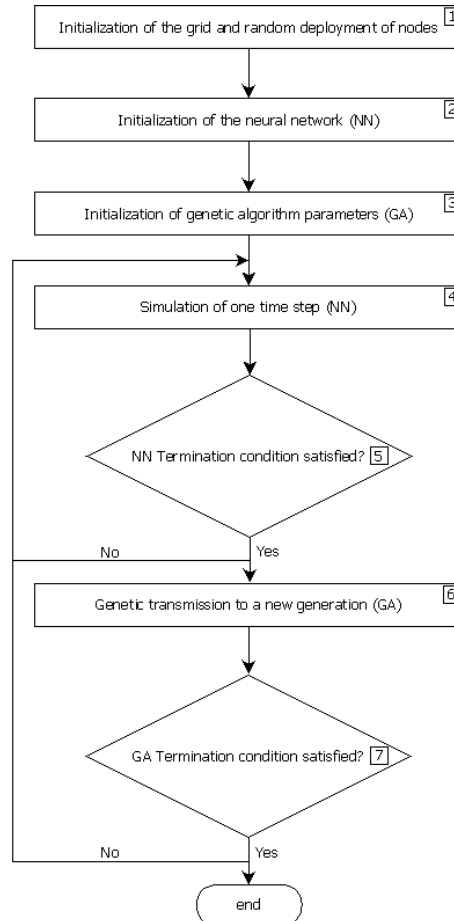


Fig. 4.11. Description of the neural network training phase.

The first assumption is necessary to express a maximum amount of distance that a node can travel in a single time step. We decided, again for sake of simplicity, to exclude four secondary directions (north-east, north-west, south-east and south-west). However, it would be very easy to extend the set of possible movements to include also the missing directions, and the algorithm would improve its performance. The second assumption allows nodes to avoid other nodes and obstacles, and to calculate the right position to move to in order not to create unwanted overlapping coverage areas with their neighbors. The boundaries of the field are considered from the nodes as obstacles. For nodes to measure the relative distances with other nodes or obstacles is possible through the measurement of the round trip time or by infrared [82], ultrasonic [83] or visual techniques [84]. In order to guarantee connectivity, we could also assume that the communication range of the nodes is at least

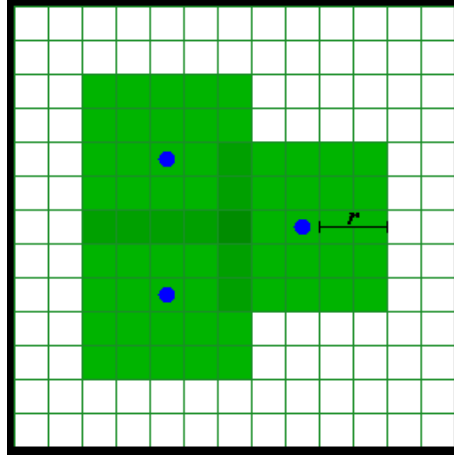


Fig. 4.12. Representation of sensing radius and cell overlapping.

twice their sensing range, as in [85], nonetheless this assumption is not strictly necessary for the coverage algorithm. When the mentioned assumptions are satisfied, no additional hardware or software is needed onboard the nodes, and also no localization systems are required. Basically, nodes are able to place themselves and maximize the coverage only with the knowledge of the local environment around themselves.

Neural Network (NN)

The behavior of each node is controlled by a fully connected, recurrent and time-discrete artificial neural network. The neural network is composed by input, output and hidden neurons, as classified also in Figure 4.13. The number of hidden neurons depends on the complexity of the problem, and in our case 2 hidden neurons are sufficient for our objective.

The input of the neural network is detected from the environment and is related to the goal of interest. In our case, the input for each node is constituted by:

1. the number of cells already covered by the neighbors, one input for each direction;
2. the distance from obstacles, one input for each direction;
3. the number of nodes placed in the same cell.

The first set of input has to objective to make node spread and maximize the coverage by avoiding overlapping. The second aims to allow nodes to learn how to avoid obstacles that constitute an impediment to movement, sensing and communication. The last set of input is specifically useful when many nodes are initially deployed in the same cell. Regarding the input, it is important to make the following remarks:

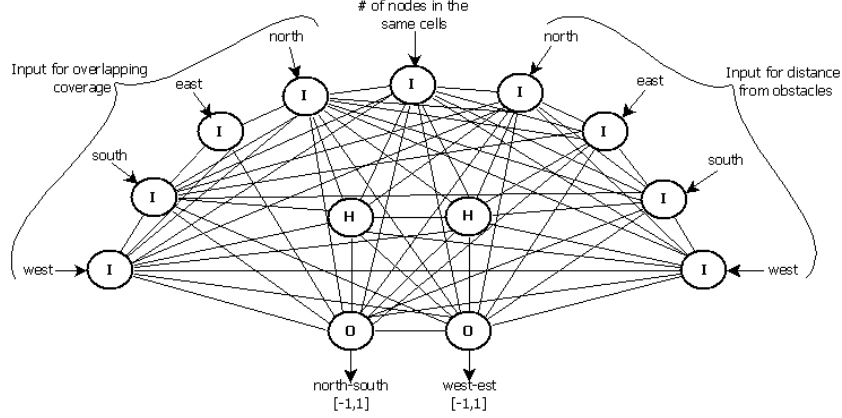


Fig. 4.13. Neural network architecture of one node for the self-deployment problem.

- all the mentioned input can be computed at each node from the measurement of the distance with other nodes (input 1 and 3) and between node and obstacles (input 2);
- the measurement is limited to the coverage range of each node.

Therefore, as mentioned in the previous subsection, only local information about relative distances is enough to feed the neural network and make the algorithm work.

In order to map the n -dimensional input into the m -dimensional output, each neuron uses a real-valued activation function and a time-varying real-valued connection with every other neuron of the network.

$$out_j(k) = F \left(\sum_{i \in N} w_{ij} \cdot out_i(k-1) + b_j \right) \quad (4.9)$$

The output of neuron j , at the time step k is the same for all the connections originating in j , therefore we indicate with out_j the output of neuron j towards all other neurons of the network. The output of neuron j is computed as in (4.9), where N is the set of all the neurons of the neural network, w_{ij} is the weight of the incoming connection from neuron i to neuron j and b_j is the bias of neuron j . Weights can have either an excitatory or inhibitory effect. In our case, the activation function F is a simple linear threshold function as expressed in equation 4.10:

$$F(x) = \begin{cases} -1.0 & \text{if } x \leq -1.0 \\ x & \text{if } -1.0 < x < 1.0 \\ 1.0 & \text{if } x \geq 1.0 \end{cases} \quad (4.10)$$

The output of the two output neurons is the output of the neural network for each node. It is composed of two real numbers that can vary in the interval

$[-1 \div 1]$, as it is clear from (4.10). Depending on these two real values, the node decides the action to make. The set of possible actions is limited to move of one cell in one of the four admitted directions or stay still in the current cell.

The whole process described above is executed in one time step, for each node of the WSRN, as shown in Step 4 of the flowchart in Figure 4.11. This phase of the training of the neural network consists in the execution of the described process until a termination condition is satisfied, as in Step 5 of the flowchart. In our case, the termination condition is either the execution of a certain fixed number of time steps or the complete coverage of the field.

Evolutionary genetic algorithm (GA) for the neural network training phase

In order to train the neural network, in a self-organizing perspective, unsupervised reinforcement learning is used. In fact, the performance of the neural network is evaluated by a fitness function.

The global optimization method used for training the neural network is a genetic algorithm. The genetic algorithm is encoded in the neural network through the genes, which are the values of the weights of the connection between each couple of neurons. A chromosome (also referred as a member of a population in the following) is constituted by an evolving set of genes, and a population is constituted by a fixed number of chromosomes. The goal of genetic algorithms is to improve the fitness function through the transmission of genes of a part of the current population from one generation to the next, as in Step 6 of the flowchart in Figure 4.11. The transmission of genes to newer generations continues until a termination condition for the genetic algorithm occurs (Step 7). In our work, the condition to terminate the genetic algorithm is the simulation of a fixed number of generations.

The transmission of genes to the new generation is based on the selection, mutation and crossover of the members of the old generation that achieved the highest value of fitness function. Thus, the fitness function has a very important role in the evolution process, since it is used as a feedback for successive generations. The objective of our work is to improve the coverage of the field. Hence, we have to relate the fitness function to the coverage achieved by each member of a population. The coverage achieved is computed by counting the distinct cells covered by all the nodes of each member of the population, upon the termination condition of Step 5 is satisfied. Thus, the fitness function could be simply represented by the number of covered cells. However, we want also to take into account the “speed” of a chromosome in reaching the maximum coverage. Therefore, we define the fitness function as follows:

$$fitness_function = \#covered_cells + (\#time_steps_max - \#time_steps_actual) \quad (4.11)$$

In (4.11), $\#time_steps_max$ is the maximum number of time steps fixed as a termination condition of Step 5 and $\#time_steps_actual$ is the actual number of time steps used in that phase of the training by the member of the population. When the difference between these two terms is not zero, it means that the neural network covered the whole field in a number of time steps lower than the maximum allowed. Therefore, an increase of the fitness function by one unit in respect of the value achieved by the previous generation can result as an effect of one the two following causes:

- one more cell has been covered in the same number of time steps;
- the same number of cells has been covered in one time step less.

Communication complexity

The proposed algorithm is based on local communication only. Indeed, a node only needs to know the position of its neighboring nodes and the surrounding obstacles to take a movement decision. In the sequel, the term “broadcast” stands for message propagation in a node’s neighborhood and the term “flooding” refers to network-wide message propagation.

After a node’s movement, an update on node’s position is broadcasted. The information contained in the broadcasted message consists of the node Id and the node position (x, y) . Here it is worth noting that, the message size is constant, therefore the message size is in $O(1)$.

Nodes will update their positions and broadcast their new information at each time steps. Since the number of time step is a constant value given as a parameter in our algorithm, this leads to a message sending complexity of $O(n)$ where n is the number of nodes. In order to select the best population to generate the next generation of our genetic algorithm, each node needs to flood the parameters of its genetic algorithm (constant number of parameters) and the value of its fitness function (number with a constant size), the message size for the flooding is thus in $O(1)$. A flooding algorithm from the literature can be used to disseminate the information. For example, MPR [86] provides a complexity in $O(\Delta^2)$ where Δ is the maximum node degree, which is the maximum number of one-hop neighbors of a node.

4.3.3 Performance evaluation

In this section we will first introduce the simulation setup and then we will show the qualitative and quantitative results of our approach, when used to solve the problem of coverage. We are mainly interested in the coverage achieved by our scheme in respect of the achievable coverage. Nonetheless, we will investigate the mechanisms of genetic transmission and learning that lie behind the usage of the fitness function. The neural network approach does not share the basic assumption of the usual approaches, because while in the

usual approach the behavior of nodes is pre-programmed (e.g., the attraction-repulsion forces in the force-based approach), neural network makes the behavior emerge from learning and interaction of the nodes. Our first objective is to focus on this emerging behavior, its effectiveness in its simplicity. Thus, a performance comparison of our scheme with a scheme using another approach would not be fair and it would be out of the scope of this work.

Simulation setup

The proposed scheme is evaluated by simulations using FREVO¹, which is an open source framework for evolutionary design. In Table 4.2, we reported all the simulation parameters used in this work. We consider a 40 cells x 40 cells field, where a variable number of nodes (n) is deployed according to a random uniform distribution and a variable number of obstacles is present (o). Also obstacles are placed according to a random uniform distribution in the field, but they can not create areas inaccessible for the nodes. Cells containing obstacles are subtracted from the achievable coverage. In our scheme, obstacles are considered impenetrable, thus limiting both nodes movement and coverage. In fact, all the cells in the coverable area of a node that are shadowed by an obstacle are not considered covered. The shadow area depends on the sensing range of the node. From some simple calculations, we can derive an overestimation of the average number of cells shadowed (cs) when an obstacle is present in the coverage area of a node:

$$cs = \frac{\sum_{l=0}^{r-1} (l+1)m(r-l)^2}{4r(r+1)} \quad (4.12)$$

where m is the number of cells in the first frame of cells around the node (8 when cells are square). If we assume $r = 2$, from (4.12) we can see that the presence of an obstacle inside the coverage area of a node creates a shadow effect that involves, on average, 2 cells (one for the physical presence of the obstacle and one because of the shadow). This is actually an overestimated value that does not take into account the possible overlapping shadow areas of obstacle, in fact the exact average value is slightly lower (1.94 cells for each obstacle). In any case, this means that even when the number of nodes is exact to cover the whole field, there will be areas uncovered due to the shadow effect of obstacles.

For our quantitative analysis, we will take into account only the physical presence (and not the shadow effect) of the obstacles by using the term *coverable area*, which indicates the total number of cells where no obstacles are present. Before starting the real simulation, we ran some test simulations in order to determine the right number of time steps and generations needed to reach a stable coverage. These results are illustrated in the next subsection, but the values used for the simulations are reported in Table 4.2, along with

¹ <http://www.frevotool.tk>

Table 4.2. Simulation parameters

Scenario parameters	
Grid height (h)	40 cells
Grid width (w)	40 cells
Number of nodes (n)	32 ÷ 96
Sensing radius (r)	2 cells
Percentage of obstacles (o)	0 ÷ 20 %
Maximum number of time steps (#time_steps_max)	100
Number of runs (runs)	10
Neural network parameters	
Total number of neurons (N)	13
Number of input neurons (I)	9
Number of hidden neurons (H)	2
Number of output neurons (O)	2
Genetic algorithm parameters	
Population size (P)	300 chromosomes
Number of generation (g)	100 generations
Percentage of elite selection (e)	15%
Percentage of mutation (mu)	45%
Percentage of crossover (c)	30%
Percentage of randomly created offsprings (off_c)	5%
Percentage of randomly selecting an offspring from previous generation (off_s)	5%

the parameters related to the neural network setup and the genetic algorithm setup. All the results have been averaged over 10 different runs in order to respect a confidence interval of 95%.

Results

A representation of the initial random deployment of the nodes is reported in Figure 4.14a and 4.14b, the figures show the scenario with 64 nodes, no obstacles (a) and 10% of obstacles (b), respectively. The figures are also useful to understand how the discretization has been realized, how the coverage area of a node and the overlapping areas are intended, and finally how obstacles impact on movement, coverage and communication. In Figure 4.14c and 4.14d we show the snapshot of the same scenarios after a training phase of 100 generations. It is possible to appreciate how nodes have been able to learn the correct placement, in order to cover the whole field when no obstacles are present (c) and as much as possible when obstacles are present (d).

Before discussing the results about the coverage, we want to justify our choice to use 100 time steps and 100 generation in the neural network simulation and the genetic algorithm, respectively.

Figure 4.15a shows the achieved coverage (in % in respect of the achievable coverage) after 100 time steps for variable number of nodes (32, 64 and 96) and percentage of obstacles (0%, 10% and 20%). As we can see, after a few number of time steps (less than 30), nodes have learnt enough to reach a stable placement and they are not able to improve further their coverage. Only the case with 64 nodes and no obstacles shows that nodes continue learning

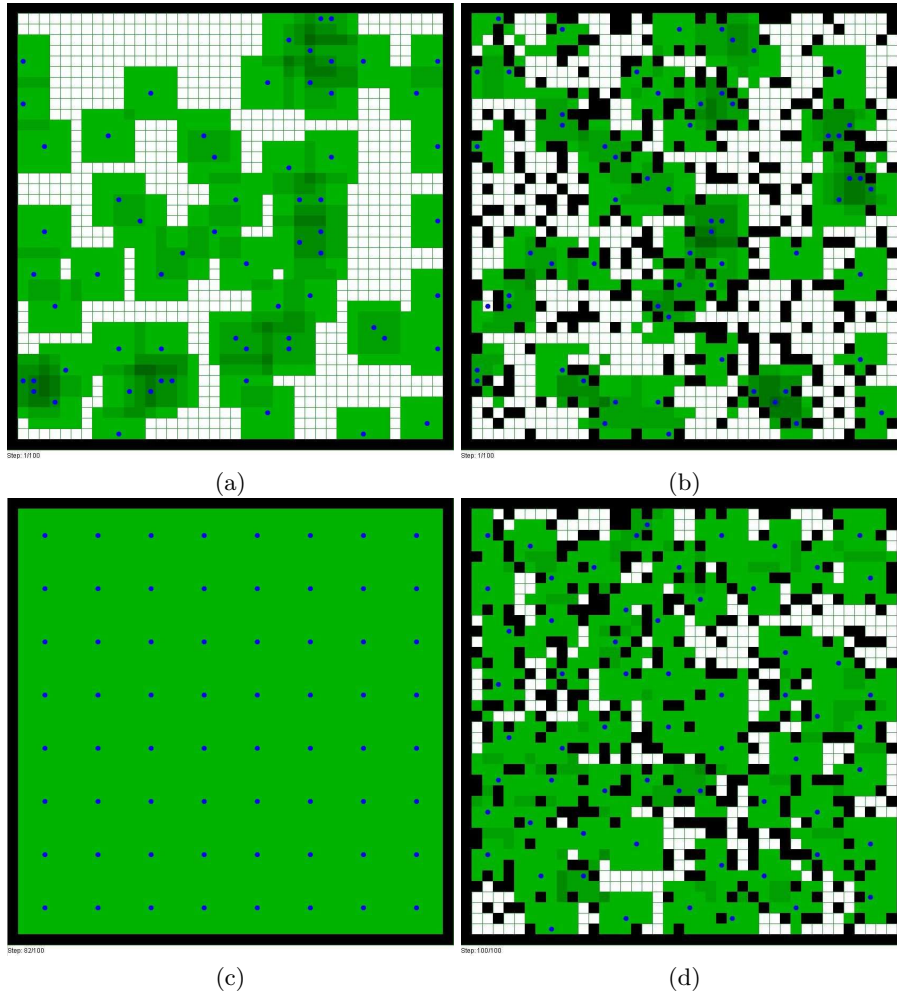


Fig. 4.14. Initial random deployment of 64 nodes ($r = 2$ cells) in a 40×40 cells grid with no obstacles (a) and with 10% of obstacles (b). WSRN self-deployment using the evolved neural network with no obstacles (c), and with 10% of obstacles (d).

till the 80th time step. In any case, 100 steps is a good value for all the simulated scenarios. In the same way, Figure 4.15b shows the value of the fitness function (that includes coverage and “speed” in covering the area) in the same aforementioned scenario. In all the simulated scenarios, the neural network is able to learn how to maximize the fitness function after about 25 generation, therefore 100 generations seems a good termination condition for the training phase.

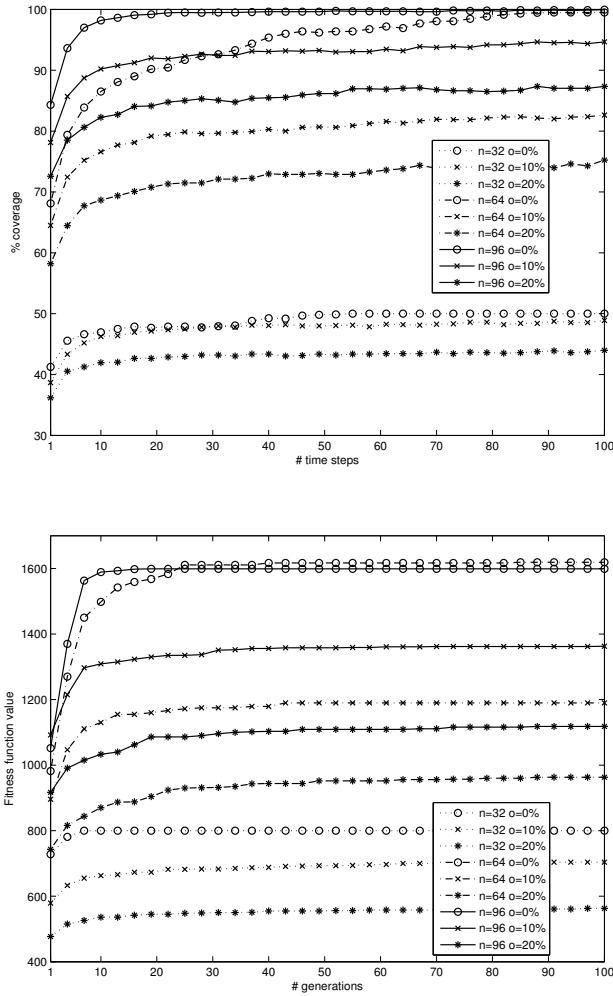


Fig. 4.15. Progress of the achieved coverage over the time by using the neural network (a). Evolution of the neural network over generations in terms of achieved fitness value (b).

From Figure 4.15b we can also appreciate other valuable information, which is the “quantity” of information that generation by generation is learnt from the nodes, and also their “speed” in learning new information. In order to give a quantitative idea of these two information, we defined the index IL , which stands for incremental learning, as follows:

$$IL = \frac{fitness(g) - fitness(1)}{fitness(100)} \quad (4.13)$$

where $fitness(x)$ is the value of the fitness function at the generation x . In Table 4.3, we reported the values of IL for all the simulated scenario, when the generation increases. We can observe that when the number of nodes is enough to cover the whole field (64 and 96 nodes), an increase in the percentage of obstacles affects the capacity to learn. This is not the case when the number of nodes is low (32 nodes) and they have more space to move without interfering with each other. When no obstacles are present, the configuration with 64 nodes is the “optimal” to cover the whole simulated field with no overlapping areas. In the Table we can see that in the configuration with 64 nodes, nodes are able to learn more and quicker than the corresponding configurations with 32 and 96 nodes. This means that an “optimal” number of nodes turns into an optimal number of learners. Improvements in the learning process would not come from an increase in the number of nodes beyond the optimal value but by an evolutionary step, such as a new input for the neural network on the nodes.

Table 4.3. Incremental learning generation-by-generation for different scenarios

g	n=32 o=0	n=32 o=10	n=32 o=20	n=64 o=0	n=64 o=10	n=64 o=20	n=96 o=0	n=96 o=10	n=96 o=20
5	0.06625	0.08665	0.07993	0.24768	0.15798	0.09034	0.29268	0.11959	0.07245
10	0.09000	0.11932	0.10480	0.31872	0.19664	0.13188	0.33583	0.15921	0.10376
20	0.09000	0.14631	0.12078	0.36195	0.22185	0.17342	0.34209	0.17461	0.15116
30	0.09000	0.14773	0.12966	0.38851	0.23445	0.19626	0.34209	0.19002	0.16011
40	0.09000	0.15483	0.13854	0.39222	0.23782	0.20872	0.34209	0.19369	0.16637
50	0.09000	0.16193	0.13854	0.39222	0.24706	0.21703	0.34209	0.19516	0.17174
60	0.09000	0.16761	0.14387	0.39222	0.24706	0.21703	0.34209	0.19736	0.17174
70	0.09000	0.17188	0.14387	0.39222	0.24706	0.22118	0.34209	0.19809	0.17352
80	0.09000	0.17188	0.14565	0.39222	0.24706	0.22534	0.34209	0.19809	0.17800
90	0.09000	0.17614	0.14920	0.39345	0.24706	0.22845	0.34209	0.19809	0.17979
100	0.09000	0.17756	0.15275	0.39345	0.24706	0.22845	0.34209	0.19883	0.17979

Figure 4.16a shows the percentage of coverage achieved calculated in respect of the coverable area. The neural network approach is compared with the maximum achievable coverage and the coverage achieved by a random deployment, which we consider as the best and the worst case, respectively. The figure is drawn for 10% of obstacles when the number of nodes increases. From Figure 4.16a, we can see that for very low and very high number of nodes the curve of the neural network approach and the maximum achievable coverage are very close to each other. For intermediate number of nodes the two curves are more distant, in particular for 64 nodes we have both the best improvement in respect of the random deployment and the highest distance from the maximum achievable. This behavior is even more evident in Figure 4.16b, where we plotted the coverage efficiency of a single node, in terms of average number of cells covered by a node. Also in this figure the neural network approach, the random deployment and the maximum achievable, are plotted when the node number increases. Note that the average value of maximum achievable coverage per node is constantly 22.5 cells (and not 25, because 10% of obstacles is uniformly distributed on the field) for number of

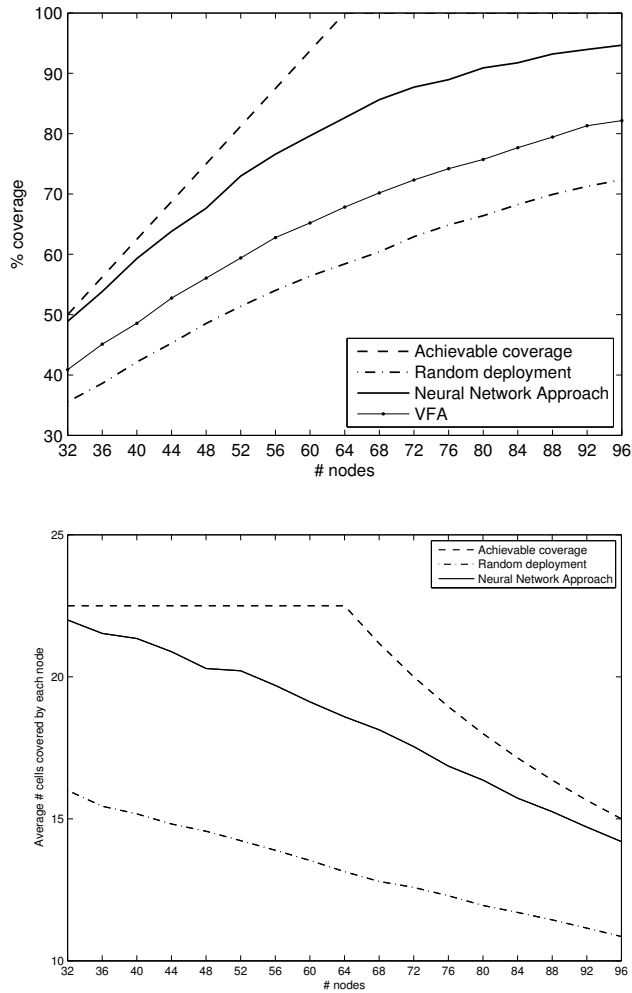


Fig. 4.16. Comparison of achieved coverage percentage (a) and nodes coverage efficiency (b) between the neural network approach, the random deployment and the maximum achievable coverage, when the number of nodes increases and 10% of obstacles are present.

nodes lower than 64, and then it decreases more than linearly, whereas the coverage efficiency for the neural network approach decreases linearly in the interval considered.

For all the aforementioned considerations about the configuration with 64 nodes, and because it is the exact number of nodes to cover the whole considered area without overlapping, we decided to analyse its performance more

in detail. In Figure 4.17 we plot the achieved coverage in respect of the percentage of obstacles present on the field for both the neural network approach and the random deployment. We have not plotted the achievable coverage, because in the scenarios with 64 nodes the whole area (100%) is coverable. Obviously, in the presence of obstacle, our neural network has the possibility to exploit the learning capabilities and outperform the random deployment approach. The interesting consideration is that for an increasing percentage of obstacles, the neural network approach decreases its performance more than linearly. As we already mentioned, the presence of a one-cell obstacle counts almost as two cells lost in the coverage, because of the physical impediment and the shadow effect, therefore a high percentage of obstacles strongly limits the evolutionary capabilities of the nodes. The only solution to change the slope of the curve in Figure 4.17 is to allow nodes to make an evolutionary step by including more input related to the presence of obstacles.

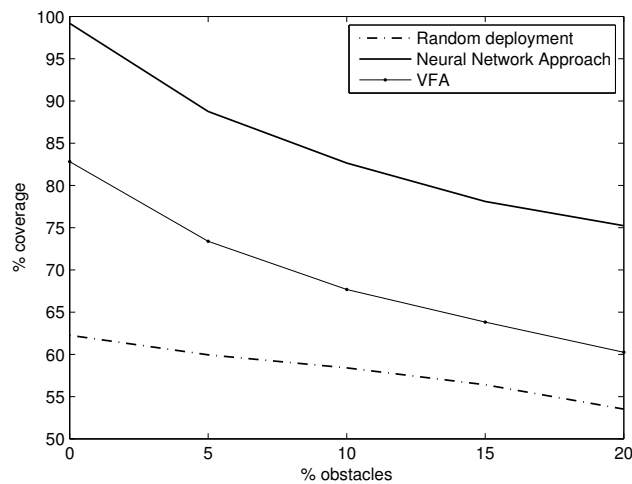


Fig. 4.17. Comparison of achieved coverage percentage between the neural network approach and the random deployment, when the number of nodes is fixed to 64 and the percentage of obstacles varies.

Finally, we want to understand the effect of the time in the learning process of the nodes. We designed the fitness function in a way that would award quick populations to transmit their genes to successive generations. Figure 4.18a shows the time step when a given coverage threshold is achieved by the population of different generations. The scenario is still 64 nodes and 10% of obstacles. We can see that, when the generation increases, higher percentage of field are covered by the nodes, and, above all, that the same percentage of coverage is achieved in a shorter time. This is exactly the behavior we tried to

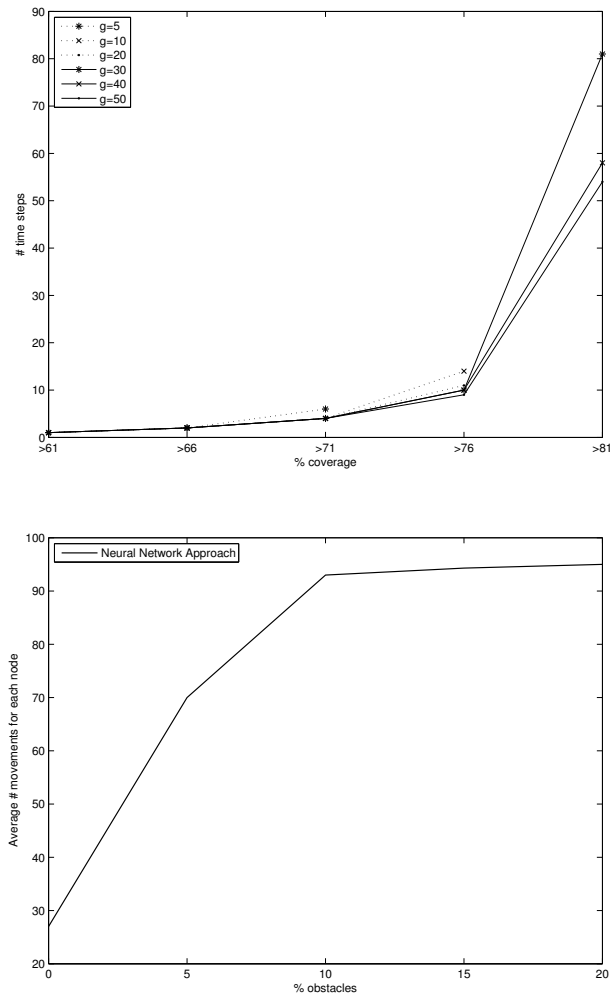


Fig. 4.18. Time step in which a given coverage percentage threshold is reached for the scenario with 64 nodes and 10% of obstacles (a), Average number of movements needed by each node to reach a stable coverage, when the number of nodes is fixed to 64 and the percentage of obstacles varies (b).

impress into population genes by including the time into the fitness function. Since we claim that the proposed neural network approach is feasible and applicable to a real case, we also investigated the number of nodes movements needed to reach the expected coverage. In Figure 4.18b, we plotted the movements needed in the scenario with 64 nodes and an increasing percentage of obstacle. The number of movements gives a measure of the energy needed for

the deployment. In our scheme only one movement is admitted for each time step. Thus, the energy consumption is implicitly reduced when the the number of time steps needed to reach a certain coverage is reduced, by exploiting the learning capabilities of the nodes.

4.3.4 Conclusion and future direction

In this section an algorithm for nodes self-deployment aimed to maximize coverage in WSRN has been proposed. The algorithm is based on neural networks and genetic algorithms. Results show how the evolving neural network approach is suitable to solve the given problem even in presence of obstacle. The approach used represents a synthesis of the most usual approaches, because it uses concepts belonging to all the three most important categories of movement-assisted deployment algorithms. But, while in the usual approaches, the behavior of the nodes is pre-programmed and can lead to unwanted situations, in the neural network approach the behavior emerges from the learning process and the interactions with the surrounding environment. The proposed algorithm achieves an high coverage of the field while minimizing the time steps needed, and consequently the number of movements and the energy consumption. The most interesting observation is that the same approach can be used, by introducing few modifications, to solve different problems and pursue different objectives. Future works can include: the definition of new input for the neural network in order to allow an evolutionary step and improve the coverage in presence of a massive quantity of obstacle; the determination of a new neural network for the implementation of another self-* property for autonomous WSRNs, and the design of new algorithms for several simultaneous objectives.

4.4 Evolving neural networks for self-control mobility to address coverage problem

An Unmanned Aerial Vehicle (UAV) is an aerial vehicle without any human operator on board that can fly autonomously or be controlled remotely. In this work we aim at a group of small, cost-efficient UAVs to achieve a common goal with better performance and/or lower cost than a system of few centrally controlled powerful but expensive UAVs. Due to unpredictability of atmospheric conditions (wind, rain, etc.), to the uncertainty of the environment (presence of obstacles, interferences, etc.) and to the dynamism of the topology of the UAV network, the setup of a predefined mobility path to follow is not feasible and due to limited bandwidth, wireless control from ground station becomes difficult, especially in presence of several UAVs. Moreover in such scenario mobility heavily impacts network performance. Consequently, an efficient mobility pattern in respect to network performance parameters is required. In this section we focus on coverage as the primary metric that

provide indication about quality of service in mobile ad-hoc networks [70]. Furthermore, the coverage problem is strictly related to target search: the higher the achieved coverage, the higher the probability of finding a given target.

A desirable way to control the UAVs' movement under such conditions is self-organizing control that means to achieve a proper coverage without any remote control or centralized scheme or pre-programmed flight path. However, the rules for such self-organizing behavior are typically hard to find, and the effects of a local rule change are often counter-intuitive [146]. Therefore, we propose the use of an evolutionary algorithm to explore the rule space. We propose the use of a neural network for drone's mobility controller since neural networks are well-suited for evolution operators such as mutation and recombination [145]. The training of the neural networks is done by reinforcement learning (without an immediate feedback) since the fitness of a particular neural network can only be evaluated after a simulation has been run for some time.

4.4.1 Proposed Model

In this work we propose a discrete time and space model based on an evolving neural network for drone's mobility controller in order to find independent and cooperative mobility patterns for UAVs such that a given area of interest is accurately covered in an efficient way.

The area of interest is discretized and for this reason we will refer it as a grid or cells. In such grid each drone starts from an initial random position (grid's cell) and in each discrete time (step) is able to move in one of four admitted direction: north (N), south (S), east (E), west (W). A cell is admitted if it is inside the grid and obstacle free.

In such model for sake of simplicity we make the following assumptions:

- the speed is fixed and similar for all UAVs;
- there is a constant radius turns of multiples of 90 degrees;
- each drone is equipped with a collision avoidance mechanism (or different altitudes are chosen);

The behavior of each drone of the network is controlled by a fully connected, recurrent and time-discrete artificial neural network. Each neuron unit in the network has a directed connection to every other unit. Each unit has a real-valued activation. Each connection has a modifiable real-valued weight. The network is composed by a layer of input neurons and a layer of output neurons and two hidden neurons (H). Usually, the number of hidden neurons is related to the problem's complexity and to the neural network expressiveness. The neural network is responsible of the mapping between the n-dimensional input and the m-dimensional output where the input of the networks is detected from the environment and is related to the goal of interest.

Independent model

In order to define a neural network for independent drones mobility the only information needed for the learning process is related to the environment. No information is acquired directly or indirectly by other drones. Focusing on our goal: maximization of the coverage, a useful input is if the movement towards each of four possible directions is admitted or not, due to the presence of an obstacle or a border. Such an input is requested from drones to have a spatial cognition and learn how to navigate the area.

The architecture of proposed neural network is shown in Figure 4.19. Four inputs are requested for obstacles presence correspondent to four direction, two outputs are needed as actuator to move in west-east or north-south direction and two hidden neuron are used.

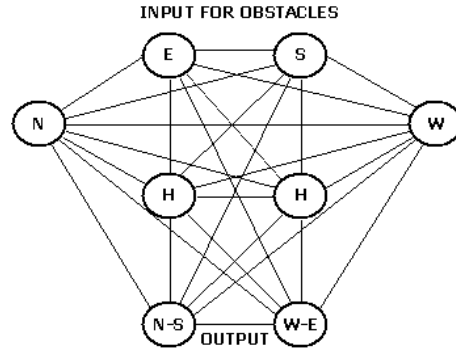


Fig. 4.19. Neural network to for independent drone’s mobility controller

Through the weighted neural network connections the inputs are mapped into the output that is simply the movement toward the new position for the node chosen among the four admitted neighbor cells. In particular at each step, each neuron i computes the sum over the current output of the neurons j feeding the connection weighted by the factor w_{ji} of the incoming connection and its bias b_i (see eq. 4.14). Weights can have either an excitatory or inhibitory effect.

$$o_i(k + 1) = F\left(\sum_{j=0}^n w_{ji}o_j(k) + b_i\right) \tag{4.14}$$

The output of the neuron for next step $k + 1$ is calculated through an activation function F over the weighted sum. In our case F is a simple linear threshold function as expressed in equation 4.15:

$$F(x) = \begin{cases} -1.0 & \text{if } x \leq -1.0 \\ x & \text{if } -1.0 < x < 1.0 \\ 1.0 & \text{if } x \geq 1.0 \end{cases} \tag{4.15}$$

Summarizing, given an input for the network the output is a movement towards a direction that could increase the achieved coverage, in this sense the neural network is the mobility controller of the drones.

Cooperative model

As many works has been shown, the cooperation can increase the performance of a network especially when the goal is shared among the components of the network. In the problem of coverage of a given area, the exchange of information among drones regarding the already covered area by each ones, could increase the coverage performance and minimize the time needed to the full area coverage. In particular, in our model, when two or more drones meet (for our discretized grid means that they are in the same cell) they are aware that a meeting is occurred and consequently the neural network get some inputs regarding the meeting. The input for the meeting is defined in a way that the neural network is able to learn the follows information:

- how many drones have met;
- from which direction each meeting is occurred.

The architecture of this new neural network is shown in Figure 4.20 and it is very similar to previous neural network for independent model the only difference is in four more input for detecting how many drones are met in each of four directions.

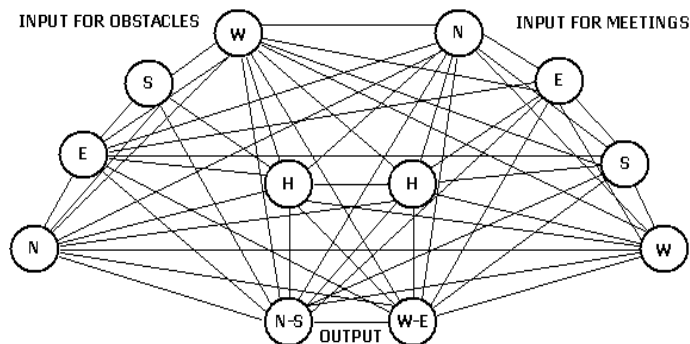


Fig. 4.20. Neural network to for cooperative drone’s mobility controller

All neural network computations to map inputs into outputs are done using equations (4.14) and (4.15) as for the independent model.

Evolutionary algorithm for training the neural network

In order to train the network, in a self-organizing perspective, reinforcement learning is used. Instead of a supervisor a fitness function is provided to evaluate the neural network’s performance.

The global optimization method used for training the neural network is a genetic algorithms and it is based on the one presented in [152]. The genetic algorithm is encoded with the neural network weights in a predefined manner where one gene in the chromosome represents one weight link or a bias of a neuron. There are many chromosomes that make up the population, therefore, many different neural networks are evolved until a stopping criterion is satisfied as in our case the maximum number of training generations has been reached. The goal of the genetic algorithm is to maximize the fitness function that is evaluated during the training phase and influences the genetic selection process.

For our problem the fitness function is defined in the simplest way to take in account the achieved coverage in terms of number of covered cells and the time needed for that coverage by the following equation:

$$fitness_function = \#covered_cells - time \quad (4.16)$$

At each generation the fitness function in equation (4.16) is evaluated and the new population encoding the weights and bias values of the neural network is generated by selection, mutation and crossover of the previous member of population that guarantee an high fitness function's value.

4.4.2 Performance evaluation

The proposed schemes are evaluated by simulations using FREVO tool ² and has been compared with some well-known synthetic mobility models like random-walk and random-direction. In this section will be detailed the simulations setup, afterwards, a results analysis is provided.

Simulations setup

The parameter of our interest is the coverage percentage of a given area. The input for the neural network are related to the possibles meeting of others drones and to the obstacles so the coverage is evaluated in respect of the increasing number of drones and in respect of the increasing percentage of obstacles in the area. The results are averaged on different scenarios where in each scenario the initial position of the drones and the obstacles are randomly placed. Among different algorithms the same random seed is used so the comparison is done on the same random scenarios. All parameters for the simulations are summarized in table 4.4.

Results

In this subsection will be illustrated some qualitative and quantitative results. Figures 4.21 and 4.22 show a scenario of a given discretized area of interest

² <http://www.frevotool.tk>

Table 4.4. Simulation parameters

Scenario parameters	
Grid height	40
Grid width	40
Number of drones	2÷20
Coverage area	25 cells
Percentage of obstacles	0÷20 %
Time steps	400
Number of runs	50
Neural network parameters	
Number of input neurons	8
Number of hidden layers	2
Number of output neurons	2
Genetic algorithm parameters	
Population size	300
Maximum number of generations	5000
Percentage of elite selection	15
Percentage of mutation	45
Percentage of crossover	30
Percentage of randomly created offspring	5
Percentage of randomly selecting an offspring from previous generation	5

respectively without obstacles and with 10% of obstacles in which 10 drones move in order to cover it after the evolution of the neural network. In these snapshot of the simulations the difference between independent and cooperative models doesn't emerge because it is strictly related to occurrence of a meeting that doesn't happen at time when Figures 4.21 and 4.22 are taken.

An interesting behavior emerging in the case without obstacles as deducible from Figure 4.21 is sweeping while in the case with obstacle shown in Figure 4.22 sweeping is not still the best way to cover the area.

In tables 4.5 and 4.6 there is a measure regarding the learning process for independent and cooperative schemes respectively in different scenarios. This quantity measure is calculated using the equation 4.17:

$$\frac{fitness(g) - fitness(1)}{fitness(5000) - fitness(1)} \quad (4.17)$$

Looking at this metric it is possible to extract the following information:

- the performance in a given number of generation (g);
- how fast is the process of learning.

First scenario regards the case with the lowest number of drones and 10% of obstacles. This is the case with the slowest learning process. Next three columns show the cases with no obstacles, 10% and 20% of obstacles respectively. The values in the entries show as for an obstacles free environment the drones learn quickly while the learning process became slower when the percentage of obstacles increase. Finally, the last column represents the case with higher number of drones and 10% of obstacles. Fixed the percentage of obstacles greater is the number of drones faster is the learning process especially for the cooperative scheme.

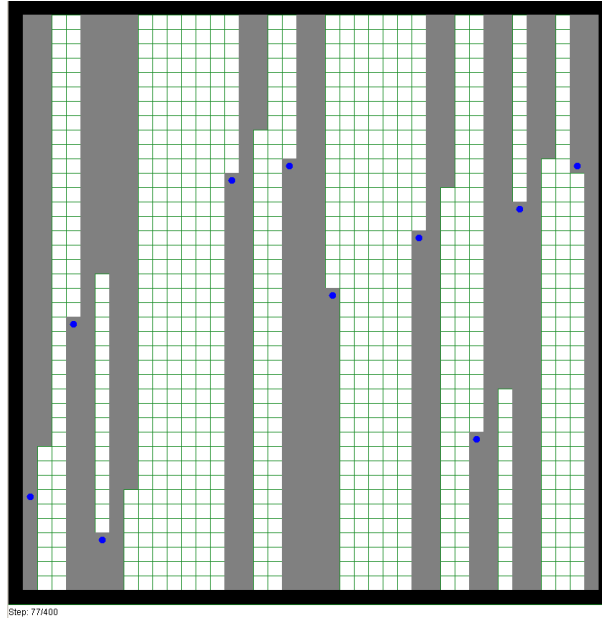


Fig. 4.21. Simulation of neural network drone's mobility controller in absence of obstacles

In order to enable the comparison with random walk and random direction, these two models are been implemented according to our discretized model that permits only four direction of movement. Such implementation is the following:

- *Random walk:* in each position, each drone can choose with the same probability the new direction among those admitted.
- *Random direction:* in each position in which the neighborhood (north, east, south and west cells) is obstacle free the drone keeps going without any change of direction, otherwise if some cell in the neighborhood is blocked the drone is free to change direction choosing with the same probability among those admitted.

For all scenarios shown in tables 4.5 and 4.6 respectively the 90% and 95% of learning process is done within the 400th generation. Figure 4.23 show the fitness function values for the same scenarios from first generation to the 400th.

Figure 4.24 shows the percentage of coverage achieved by the two different schemes: independent and cooperative. The cooperative is able to learn from the other drones and get the highest coverage. The gap between cooperative and independent is around 5% and rise to 15% in respect to the random-direction.

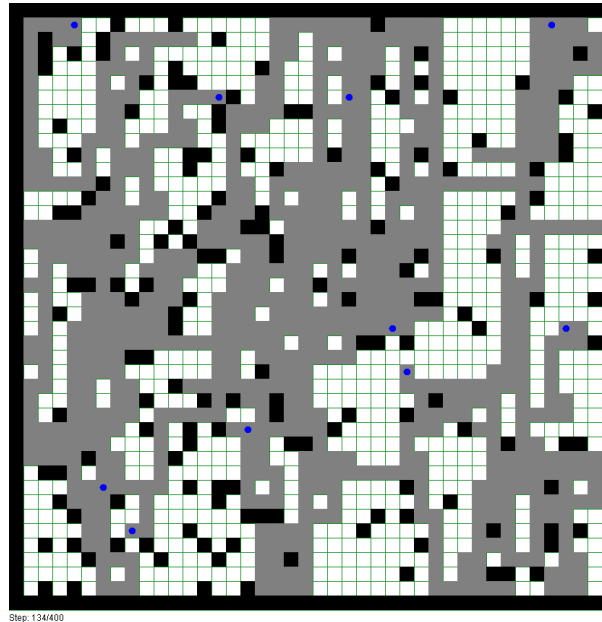


Fig. 4.22. Simulation of neural network drone’s mobility controller in presence of obstacles

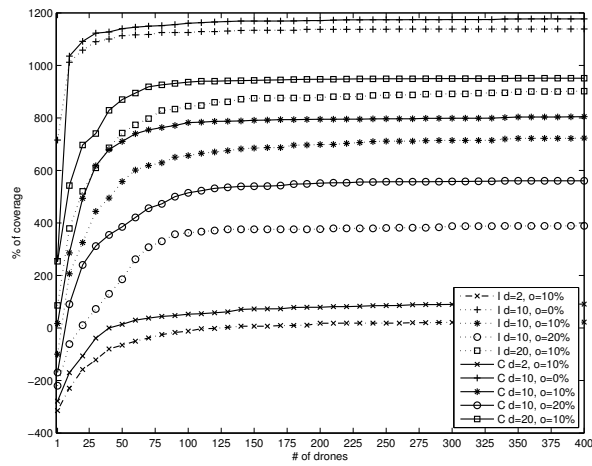


Fig. 4.23. Fitness function value with (d) drones and (o) obstacles for Cooperative (C) and Independent (I) schemes

Coverage percentage versus percentage of obstacles is shown in Figure 4.25. Neural network approach gives good results especially in absence of

Table 4.5. Cumulative learning measure generation-by-generation for Independent scheme in different scenarios

g	d=2 o=10	d=10 o=0	d=10 o=10	d=10 o=20	d=20 o=10
10	0.2301	0.6955	0.3553	0.2449	0.3543
20	0.4224	0.8045	0.4934	0.3561	0.5250
30	0.5205	0.8802	0.6312	0.4512	0.6331
40	0.6325	0.9037	0.6902	0.5401	0.7255
50	0.6713	0.9337	0.7628	0.6260	0.7935
60	0.7123	0.9408	0.8136	0.7440	0.8312
70	0.7452	0.9445	0.8338	0.8147	0.8600
80	0.7802	0.9624	0.8459	0.8496	0.8972
90	0.8023	0.9624	0.8705	0.8907	0.9040
100	0.8152	0.9624	0.8777	0.8987	0.9175
200	0.8966	0.9901	0.9267	0.9209	0.9570
300	0.9052	0.9930	0.9445	0.9382	0.9739
400	0.9084	0.9934	0.9550	0.9401	0.9862
500	0.9170	0.9944	0.9654	0.9419	0.9901
600	0.9359	0.9948	0.9654	0.9472	0.9908
700	0.9494	0.9962	0.9659	0.9611	0.9918
800	0.9531	0.9967	0.9715	0.9648	0.9918
900	0.9531	0.9967	0.9745	0.9688	0.9942
1000	0.9537	0.9967	0.9763	0.9688	0.9942
2000	0.9564	1.0000	0.9793	0.9852	0.9959
3000	0.9903	1.0000	0.9879	0.9883	0.9976
4000	0.9968	1.0000	0.9933	1.0000	0.9983
5000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 4.6. Cumulative learning measure generation-by-generation for Cooperative scheme in different scenarios

g	d=2 o=10	d=10 o=0	d=10 o=10	d=10 o=20	d=20 o=10
10	0.2788	0.8360	0.3333	0.3447	0.4073
20	0.4419	0.8970	0.5907	0.5431	0.6257
30	0.6163	0.9308	0.7441	0.6384	0.6873
40	0.7166	0.9358	0.8209	0.6946	0.8124
50	0.7521	0.9493	0.8590	0.7355	0.8701
60	0.7927	0.9555	0.8957	0.7830	0.9051
70	0.8133	0.9597	0.9138	0.8294	0.9387
80	0.8297	0.9620	0.9244	0.8517	0.9492
90	0.8374	0.9664	0.9343	0.8880	0.9568
100	0.8513	0.9716	0.9477	0.9074	0.9636
200	0.9192	0.9827	0.9636	0.9565	0.9794
300	0.9475	0.9870	0.9685	0.9647	0.9828
400	0.9501	0.9898	0.9760	0.9682	0.9847
500	0.9599	0.9898	0.9765	0.9713	0.9850
600	0.9681	0.9902	0.9814	0.9732	0.9870
700	0.9712	0.9902	0.9854	0.9756	0.9898
800	0.9794	0.9905	0.9861	0.9756	0.9898
900	0.9805	0.9905	0.9881	0.9769	0.9907
1000	0.9805	0.9924	0.9881	0.9804	0.9915
2000	0.9933	0.9950	0.9943	0.9936	0.9949
3000	0.9959	0.9967	0.9988	0.9971	0.9955
4000	1.0000	0.9987	0.9995	0.9992	0.9969
5000	1.0000	1.0000	1.0000	1.0000	1.0000

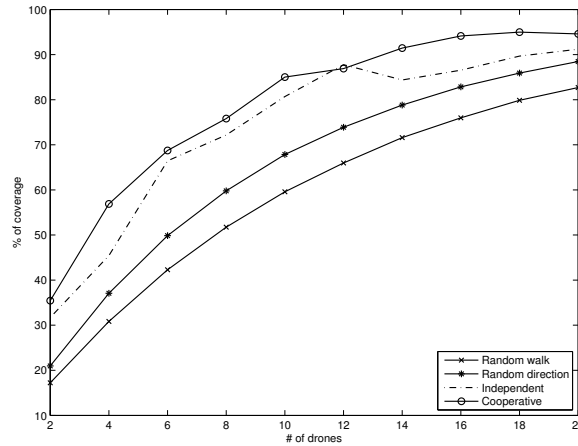


Fig. 4.24. Coverage percentage with increasing number of drones

obstacles. When the percentage of obstacles increase the achieved coverage decrease but the cooperative scheme obtain always at least a 15% of coverage more than the random direction. Also in this case the cooperative scheme outperforms the independent except for the case with 15% of obstacles. The highest difference among cooperative and independent schemes is achieved in correspondence of 20% of obstacles where the difficulties in navigation due to obstacles are smoothed thanks to collaboration among drones. The initial increasing of coverage in random-direction is probably due to the particular implementation of this mobility model adapted to our needs.

4.4.3 Conclusion

In this section a drone's mobility controller using neural network has been proposed. An evolutionary algorithm is used to train the network in order to maximize the coverage of an interest area. Two different approaches are compared: the independent approach, where the drone's neural network only receives information related to the environment (presence of obstacles), and the cooperative approach where UAVs exchange information when a meeting occurs and, in this case, each drone's neural network also receives information regarding the direction of provenience of the drones met. The proposed approaches have been compared with random-walk and random direction. The results, obtained by extensive simulation, show that the cooperative neural network approach outperform the other schemes except for the case when there is an high presence of obstacles, where the independent scheme has the best performance.

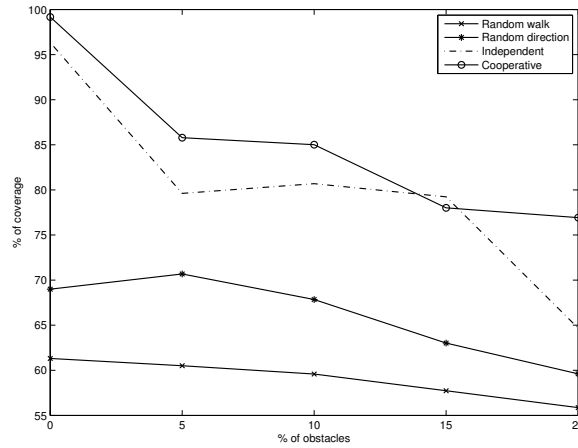


Fig. 4.25. Coverage percentage with increasing percentage of obstacles

4.5 Conclusion

In this chapter we have shown that nodes of a network, which act as a swarm, can cover a given area of interest even in case the distribution of the events is not uniform. Moreover, PSO can be used for target tracking. A similar problem has been investigated through an evolving neural network. We have shown that each node is able, after a training phase, to learn some general rules that bring out an efficient behaviour for achieving full coverage, even in presence of obstacles. A similar neural network is then used for the problem of sweeping an area by using collaborative UAVs. Also in this latter case UAVs are able to learn how to achieve a full coverage of the area in the minimum time.

A case of study: Efficient Coverage for Grid-Based Wireless Sensor Networks

5.1 Introduction

In this final chapter a case of study is analyzed where all the used methodologies are compared: optimization model, distributed heuristic and bio-inspired mechanism. In particular a genetic algorithm, previously illustrated in this thesis, is applied in order to achieve an efficient coverage for Grid-Based Wireless Sensor Network. A new coverage technique is proposed, totally distributed and based on local information, for wireless sensor networks in which the sensors are arranged in a square grid. We show how this extremely simple technique is able to reach good performance in terms of coverage even though different configurations of networks are considered. Specifically, we focus on networks with and without obstacles with different densities of nodes and we show how this coverage technique is effective. In order to demonstrate the effectiveness of our coverage technique on square grid networks, we consider an optimization model and other two coverage algorithms: the first one is based on a genetic approach and the latter is based on the well-known concept of virtual forces. Results of simulations show how in the worst case the heuristic coverage is 14 % smaller than the coverage obtained with the optimization model and the results of our approach are very close to those obtained with the virtual forces based technique. When the size of the network increases the searching space of our scheme increases too and we obtain better performance in terms of coverage. In fact, for greater network size the heuristic outperforms the virtual forces approach, both without and with obstacles. Moreover, we will show how our approach is extremely flexible and adaptive to the changes of the networks by considering different interest zones. In fact, our technique is able to detect zones that need to be monitored by more nodes and a higher concentration of nodes will be obtained in these zones.

5.2 State of art

In recent years, advanced VLSI and radio frequency (RF) technologies have accelerated the development and applications of wireless sensor and distributed networks [124], [125]. This kind of networks are starting to play an important role in a wide range of applications such as medical treatment [119], battlefield surveillance [120], habitat monitoring [121], [122] and so forth. In distributed Wireless Sensor Networks (WSNs), design of the placement of sensors devices in the sensor field is a very important issue. Efficient sensor placement strategies allow to minimize cost and reach an high level of accuracy [123]. In the art gallery problem, cameras are deployed such that the whole gallery is thief-proof [36], [135]. A wireless sensor network must achieve the specified coverage level of the application so that the quality of service provided by the wireless sensor network can be guaranteed. Many sensor deployment algorithms attempt to fully cover a sensor field using the minimum sensors or the minimum cost of sensors. In [79] authors propose a method to deploy sensors to provide full coverage on a sensor field with obstacles. In [126] authors propose a distributed algorithm in order to activate a subset of sensors to fully cover the entire sensor field at one time. In [127] authors try to ensure that each point in a sensor field is covered by at least k sensors. In order to cover in an effective way the field, they select a subset of sensors for ensuring k -coverage and k' -connectivity. Some coverage techniques consider the components in the sensor fields such as sensors, obstacles or preferential fields as virtual forces sources and try to balance virtual forces through the deployment of the devices [128] and [89]. In [129] authors propose CCAN that selects a connected dominating set in a dense wireless sensor network such that the coverage probabilities of the given points (i.e. specific targets or given points in a sensor field that need to be more monitored), are larger than a given parameter. In many applications, some specific areas need to be more monitored and in this case the problem is of constructing a wireless sensor network to fully cover critical grids. In [130] authors consider the problem of constructing a wireless sensor network to fully cover critical grids of equilateral triangles by deploying the minimum sensors on grid points and they show that this problem is NP-Complete. In the circle covering problem, usually equal circles are used to fully cover equilateral triangles [137], rectangle [136], and squares [138]. The circle covering problem is different from the covering problem considered in this section since the circles are independent and can be moved anywhere. Moreover, we have to consider the constraint of connectivity, so the potential available position in the sensor field are limited. Usually a sensor field is divided into square grids [123], [75] and in [131] the problem of deploying the minimum number of sensors on grid points to construct a wireless sensor network fully covering critical square grids is considered and authors show that this problem is NP-Complete. In this chapter we focus on a grid sensor field in which different situations may occur, i.e. a uniform coverage could be necessary and the sensor field is free in the sense that there are not obstacles.

Another application could consider a grid sensor field where obstacles occupy some grid points that can not be occupied by sensor devices. Eventually, some specific application could require that some specific critical grid points need to be more monitored than other points. In this chapter we propose a very simple heuristic only based on local information and in a totally distributed way is able to move sensor devices to the closer position in order to satisfy the specific coverage requests as those considered above. In order to show the effectiveness of this simple technique we introduce an optimization model that is able to find the best position in terms of coverage and two approaches, the first one is a centralized technique based on a genetic algorithm and the latter is a virtual forces based approach. Simulation results show how our techniques is able to obtain coverage degrees close to the optimization model and the genetic approach and when the size of the network increases our technique overcomes the virtual forces techniques both when the obstacles are considered and not. The reminder of the chapter is organized as follows. In Section 2 we consider square grids. In Section 3 we present the optimization model used as benchmark. Section 4 is to explain the genetic approach and the virtual forces technique. In Section 5 we give the detail of the simple heuristic we propose. Section 6 is to explain the results and finally, we conclude the chapter and consider a discussion of future research in Section 7.

5.3 Square Grid Networks

A wireless sensor network is said *grid-based* if it consists of a (potentially unbounded) number of identical sensors arranged in a square grid as shown in Figure 5.1. Let a sensor has a maximum transmission range said t_r , each node is able to communicate with all nodes within the circle of radius t_r in a direct way. In a grid-based network we need to model this in a different way, i.e. a *Lee sphere* of appropriate radius [132]. Let the length of the sides of the squares of our grid be 1. The distance among the squares in the grid can be measured in terms of Manhattan metric, that is the distance between two squares is the sum of the horizontal and vertical distances between the centers of the squares. With “horizontal” and “vertical” we indicate the two perpendicular directions that are parallel to the sides of the squares in the grid.

Definition 1 *A Lee sphere of radius r centered at a given square consists of the set of squares that lie at Manhattan distance at most r from that square as shown in Fig. 5.2(a) and 5.2(b)*

This approximation facilitates the representation of the coverage in the grid-based networks. In this chapter we refer to the Lee sphere of radii 2 as shown in Fig. 5.2(b). In practice, all the points that are within the squares are considered covered by a node at the center of the Lee sphere.

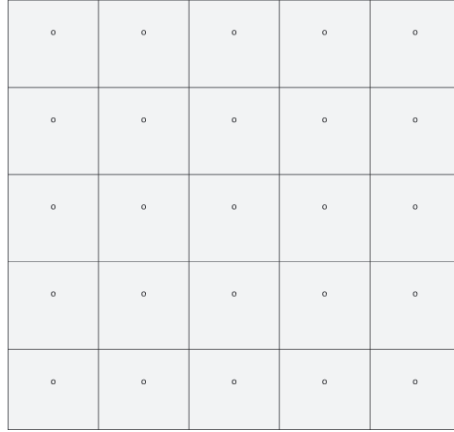


Fig. 5.1. Arrangement of sensors in a square grid.

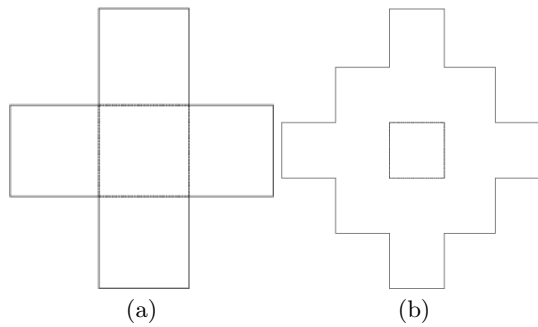


Fig. 5.2. Lee spheres of radii 1 (a) and 2 (b).

5.4 The optimization model

In this section, we give the mathematical formulation of the problem under study, as an integer nonlinear programming model. It is assumed that the sensor field is represented by a two-dimensional grid. The parameters used for the formulation are the following: h : the grid height, w : the grid width; d : the discretization step, n : the number of available sensors; r : the coverage radius; M : a large positive number.

The variables of the proposed model are: (x_k, y_k) , $k = 1, \dots, n$ the Cartesian coordinates that indicate the location of the sensor k in the sensor field; ϕ_{ijk} , $i = 1, \dots, \lceil h/d \rceil$, $j = 1, \dots, \lceil w/d \rceil$, $k = 1, \dots, n$ a binary variable that takes the value one if the location (i, j) is covered by sensor k , and zero otherwise; δ_{ij} , $i = 1, \dots, \lceil h/d \rceil$, $j = 1, \dots, \lceil w/d \rceil$, a binary variable that takes the value one if the location (i, j) is covered by at least one sensor, and zero otherwise.

The considered problem can be mathematically stated as follows:

$$\max \sum_{i=1}^{\lceil h/d \rceil} \sum_{j=1}^{\lceil w/d \rceil} \delta_{ij} \quad (5.1)$$

$$r - \sqrt{(i - x_k)^2 + (j - y_k)^2} \geq M (\phi_{ijk} - 1), \quad \forall i, j, k \quad (5.2)$$

$$\delta_{ij} \leq \sum_{k=1}^n \phi_{ijk}, \quad \forall i, j \quad (5.3)$$

$$M \delta_{ij} \geq \sum_{k=1}^n \phi_{ijk}, \quad \forall i, j \quad (5.4)$$

$$0 \leq x_k \leq \lceil h/d \rceil, \quad 0 \leq y_k \leq \lceil w/d \rceil, \quad \forall k \quad (5.5)$$

$$x_k, y_k \text{ integer}, \quad \forall k \quad (5.6)$$

$$\phi_{ijk} \text{ binary}, \quad \forall i, j, k \quad (5.7)$$

$$\delta_{ij} \text{ binary}, \quad \forall i, j \quad (5.8)$$

The objective function in (5.19) maximizes the number of locations covered by at least one sensor. Conditions (5.2) state that if the euclidean distance between the sensor k and the location (i, j) is lower than or equal to the coverage radius r than the variable ϕ_{ijk} takes the value one, otherwise it is set to zero. Constraints (5.3) and (5.4) are logical constraints and ensure that the indicator variable δ_{ij} takes on a value of one if the location (i, j) is covered by at least one sensor and zero otherwise. Finally, conditions (5.5)-(5.8) represent the variable domain constraints.

The mathematical formulation reported above is an integer nonlinear programming model, where the nonlinearity is confined to the constraints (5.2).

In order to derive an integer linear model, the euclidean distance has been replaced by the following expression: $d_{x_{ik}} + d_{y_{jk}} - 0.5 \min(d_{x_{ik}}, d_{y_{jk}})$, where $d_{x_{ik}} = |i - x_k|$ and $d_{y_{jk}} = |j - y_k|$. This approximation, introduced in [133], overestimates distance and yields error in interval 0% to 12%.

To eliminate the terms with the absolute value, we introduce the additional constraints reported below ([134]):

$$d_{x_{ik}} \geq i - x_k \quad \forall i, k \quad (5.9)$$

$$d_{x_{ik}} \geq -i + x_k \quad \forall i, k \quad (5.10)$$

$$d_{y_{jk}} \geq j - y_k \quad \forall j, k \quad (5.11)$$

$$d_{y_{jk}} \geq -j + y_k \quad \forall j, k \quad (5.12)$$

Thus, constraints (5.2) are replaced by the following conditions:

$$r - d_{x_{ik}} + d_{y_{jk}} - 0.5 \min_{ijk} \geq M (\phi_{ijk} - 1), \quad \forall i, j, k \quad (5.13)$$

where the variables \min_{ijk} , $\forall i, j, k$ take the minimum value between $d_{x_{ik}}$ and $d_{y_{jk}}$. Consequently, the following constraints need to be satisfied:

$$\min_{ijk} \leq d_{x_{ik}} \quad \forall i, j, k \quad (5.14)$$

$$\min_{ijk} \leq d_{y_{jk}} \quad \forall i, j, k \quad (5.15)$$

The proposed integer linear model optimizes the function (5.19) subject to constraints (5.9)-(5.15) and (5.3)-(5.8).

It is worth observing that the mathematical formulation reported above can be easily extended to handle the specific situation in which some obstacles are present in the sensor field.

In this particular case, it is sufficient to impose that the locations occupied by the obstacles cannot be considered as feasible positions where the sensors can be localized.

Let $\mathcal{OBS} = \{(\tilde{i}, \tilde{j}) \mid \tilde{i} \geq 1 \text{ and } \tilde{i} \leq \lceil h/d \rceil \text{ and } \tilde{j} \geq 1 \text{ and } \tilde{j} \leq \lceil w/h \rceil\}$ be the set of the positions in which the obstacles are located, the mathematical formulation of the problem in which the obstacles are considered, is obtained by replacing in the model reported above the constraints (5.5) with the following conditions:

$$0 \leq x_k \leq \lceil h/d \rceil, \quad 0 \leq y_k \leq \lceil w/d \rceil, \quad (x_k, y_k) \notin \mathcal{OBS} \quad \forall k \quad (5.16)$$

In some applications, there are the needs to guarantee that some zones (the so-called zones of interest) are covered up to a certain degree.

The proposed mathematical models can be easily extended to handle also this specific situation. In particular, let $cover_{ij}$ denote the minimum number of sensors that have to cover the location (i, j) , $i = 1, \dots, \lceil h/d \rceil$, $j = 1, \dots, \lceil w/h \rceil$, the problem under consideration can be represented mathematically by the models introduced above in which the following constraints are added:

$$\sum_{k=1}^n \phi_{ijk} \geq cover_{i,j}, \quad \forall i, j \quad (5.17)$$

It is worth observing that an alternative way to address the problem in which zones of interest are specified is to maximize the number of sensors

that cover these zones, by ensuring that each location is covered by at least one sensor. In particular, let \mathcal{ZOI} denote the set of locations corresponding to the zone of interests, the objective function to be maximized assumes the following form:

$$\sum_{k=1}^n \sum_{(\tilde{i}, \tilde{j}) \in \mathcal{ZOI}} \phi_{\tilde{i}\tilde{j}k} \quad (5.18)$$

subject to constraints (5.3)-(5.15) and the condition reported in what follows:

$$\sum_{i=1}^{\lceil h/d \rceil} \sum_{j=1}^{\lceil w/d \rceil} \delta_{ij} \geq \gamma \quad (5.19)$$

where γ is set to $\lceil h/d \rceil \times \lceil w/d \rceil$, if each location has to be covered by at least one sensor. It is important to observe that it is possible to carry out a sensitive analysis on the value of the parameter γ in order to find the best trade-off between the maximization of number of sensors that should cover the locations belonging to \mathcal{ZOI} and the ensuring of an acceptable coverage for the remaining locations.

5.5 Coverage Techniques

In this section we describe two different coverage techniques: the first one based on a genetic approach and the latter based on the virtual forces approach. All the techniques considered in this chapter are based on the grid-based model described in Section 2 as far as the sensing model is concerned. As already outlined, we consider Lee sphere of radii 2.

5.5.1 Genetic Algorithm

The proposed genetic algorithm is used as optimization method in order to find an optimal solution reducing the solution space through this evolutionary approach. A particular placement is coded as a member of a population of the genetic algorithm, such representation is given by the sequence of x and y coordinates of all sensors in the grid as illustrated in Figure 5.3.

x_1	y_1	x_2	y_2	\dots	x_n	y_n
-------	-------	-------	-------	---------	-------	-------

Fig. 5.3. Placement coded in a chromosome

The optimization problem is formalized as the minimization of uncovered cells, consequentially, the fitness function used for the evolution of the population is the following:

$$f = \#_total_cells - \#_covered_cells \quad (5.20)$$

This function takes as an input a given placement and, where specified, also the positions of the obstacles on the grid and gives as output a fitness value that represents the number of holes in the grid. Substantially, the population evolves through the genetic algorithm using the fitness function as supervisor of the evolution process.

The steps of the algorithm are the following:

1. random initialization of a population;
2. ranking of population members according to the fitness function;
3. selection, mutation and crossover of best members of populations;
4. repeat from point (1) until termination condition (maximum number of generations) is satisfied.

5.5.2 Virtual Forces Algorithm (VFA)

VFA is a well-known technique used for deployment in wireless sensor networks and allow to reach good performance in terms of coverage from a random placement using a weighted combination of repulsive and attractive forces based on mutual distances among nodes. Pros of this approach is the flexibility in adapting in different scenarios also in presence of obstacle or zones of interest that are considered as repulsive and attractive forces respectively. Cons regard the difficult to tune opportunely some algorithm's parameters like ω_a and ω_r that depends on particular scenario and the nature of the algorithm that is centralized and in particular is based on clustering so it has all disadvantages of having a central unit of elaboration.

5.5.3 A Map-assisted coverage heuristic

In this section we give the details about the coverage map assisted technique. It is worth to notice that this technique is totally distributed and based on local information.

The proposed algorithm is based on a local coverage grid map as depicted in Figure 5.4. It is worth observing that a node can exploit information that are within the secondary frame (we consider a transmission range greater than the sensing range) even if the coverage is computed by considering the *Lee sphere* model as shown in Fig. 5.2 (b). Each node, communicating with neighbors is able to calculate the partial map map needed to compute the movement towards the right direction. The black cells are the location occupied by nodes while the number in each free cell is the degree of coverage of that cell (number of nodes that cover the cell). Of course, the number in the cell could also acquire different meanings as the degree of interest of a specific point and/or zone.

The basic idea of the algorithm is to make move the nodes toward the direction that seems less covered looking only at the map in correspondence

0	1	0	0	1	1	1	2	1	0
1	1	1	1	0	2	2	2	1	1
1	1	2	2	2	2	3	1	2	2
1	2	2	2	3	2	2	3	2	1
1	1	1	2	3	2	1	1	2	1
1	1	1	2	1	2	1	1	1	2
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	1	1	1
1	1	1	1	0	0	0	0	1	0
1	1	1	1	1	0	0	0	0	0

Fig. 5.4. Coverage grid example with $r = 2$, the gray cells indicate the presence of a sensor in the specific cell.

of the cells at the borders of coverage area of each node. In particular given the sum over $(2r + 1)$ cells, on eight area coverage borders (North (N), North-East(NE), East (E), South-East (SE), South (S), South-West (SW), West(W), North-West (NW)) each node moves toward the direction opposite to the direction with the higher coverage value. In Figure 5.4 in order to figure out how the borders are considered the North and South-East borders are highlighted in red. The algorithm considers the information as shown through the blue frame.

The details of the algorithm are given in the following pseudo-code:

Algorithm 4 Coverage Map based Movement Algorithm

(x, y) : current position of node;

r : coverage radius;

repeat

 calculate map;

 set $N_coverage = \sum_{j=x-r}^{x+r} map(y+r, j)$;

 set $E_coverage = \sum_{i=y-r}^{y+r} map(i, x+r)$;

 set $S_coverage = \sum_{j=x-r}^{x+r} map(y-r, j)$;

 set $W_coverage = \sum_{i=y-r}^{y+r} map(i, x-r)$;

 set $NE_coverage = \sum_{j=x}^{x+r} map(y+r, j) + \sum_{i=y+r}^y map(i, x+r)$;

 set $SE_coverage = \sum_{j=x}^{x+r} map(y-r, j) + \sum_{i=y-r}^y map(i, x+r)$;

 set $SW_coverage = \sum_{j=x-r}^x map(y-r, j) + \sum_{i=y-r}^y map(i, x-r)$;

 set $NW_coverage = \sum_{j=x-r}^x map(y+r, j) + \sum_{i=y}^{y+r} map(i, x-r)$;

 set $N - S = N_coverage - south_coverage$;

 set $NE - SW = NE_coverage - SW_coverage$;

 set $W - E = N_coverage - S_coverage$;

 set $SE - NW = SE_coverage - NW_coverage$;

 set $max_difference = \max(|N - S|, |NE - SW|, |W - E|, |SE - NW|)$;

 move $\frac{|max_difference|}{(2r+1)}$ step choosing in a equiprobable way among one of $maximum_difference$ direction standing still if all direction obtain the same value;

until $(x(t), y(t)) == (x(t-2), y(t-2))$ or $(x(t), y(t)) == (x(t-3), y(t-3))$ or $(x(t), y(t)) == (x(t-4), y(t-4))$

5.6 Performance Evaluations

In this section, we evaluate the performance of the proposed heuristic and we compare our technique with a centralized coverage algorithm based on a genetic approach and a virtual forces based approach. In Fig. 5.5 we show percentage of coverage achieved with the 4 techniques previously described. It is worthwhile to notice how an increased size of the network corresponds to a better behavior of the Heuristic. In fact, when we compare coverage achievability obtained with our technique and the virtual forces approach, we can observe how the heuristic outperforms the virtual force technique when size of the networks increases. This result is encouraging because it implies that our heuristic is scalable. This better behavior is related to an increased movement space. In practice, with the 15x15 grid, the heuristic can exploit more information and for that results are better. These considerations are confirmed when we consider a 25x25 size of the grid as shown in 5.5 (c).

It is worth to notice that for 25x25 sensor field size we reported results of neither the Optimization Model nor the Genetic Algorithm (GA). The reason is that when the size of the networks and the number of nodes increase, the number of potential solution to be evaluated increase too and both the approaches are extremely inefficient and time-consuming.

In 5.6(a) and 5.6(b) we show the results of the 4 techniques when in the grid are present a certain percentage of obstacles. In this chapter we refer to obstacle as something that occupies either a part of a square or a whole square. Specifically, we consider a number of obstacles ranging from 0 to 20 % in respect of the number of squares of the field. In this case the coverage is computed by eliminating the squares occupied by the obstacles. The optimization model and the genetic approach are able to reach the maximum achievable coverage. This is due to the fact that both, the optimization model and the genetic approach are based on global information and know perfectly the positions already occupied that can not be occupied by others sensor devices. Also in the case we can observe how the heuristic works well when the number of information that can exploit increases. In fact, also in this case the achieved coverage is higher when the size of the grid is 15x15 and increases more when the 25x25 size of the grid is considered.

5.6.1 Zone of Interest

In this subsection we introduce the concept of Zone of Interest (ZoI). In a realistic sensor field covered by a wireless sensor network it could happen that some points is “more” interesting than other points and need to be “more” monitored than others points. Specifically, we show this situation in Fig. 5.7, where we show a specific point that could be characterized with different “degree” of interests. Darker is the color higher is the interest of the specific zone and consequently more nodes are needed in the specific point.

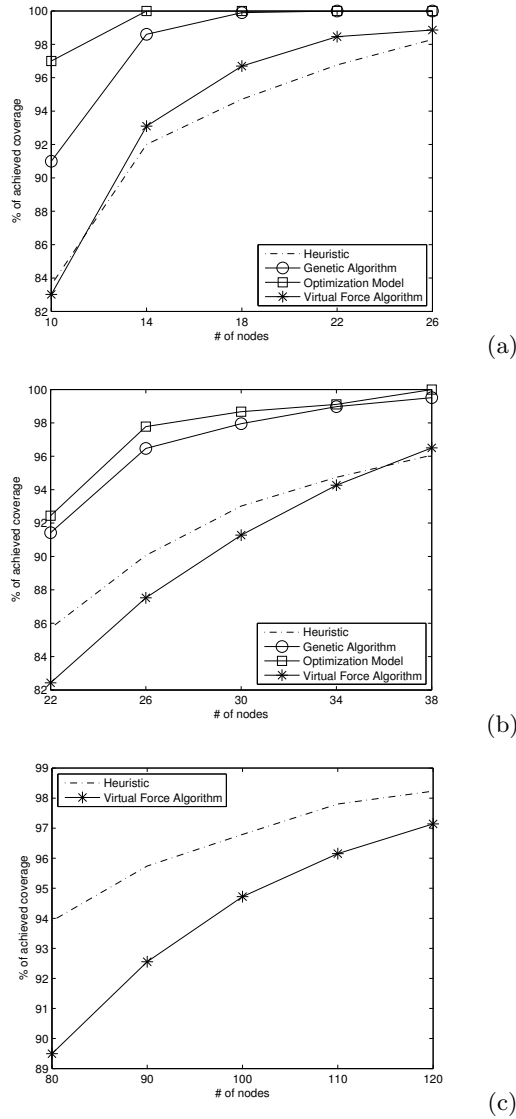


Fig. 5.5. Percentage of coverage when the grid size is 10x10 (a) 15x15 (b) and 25x25 (c).

The “degree” of interest of either a certain zone or a point in a field can be defined in different ways, i.e. we can explicitly declare the number of nodes we need to cover a specific point/zone such as in the Optimization Model defined above or we can simply define a certain degree of interest through a weight

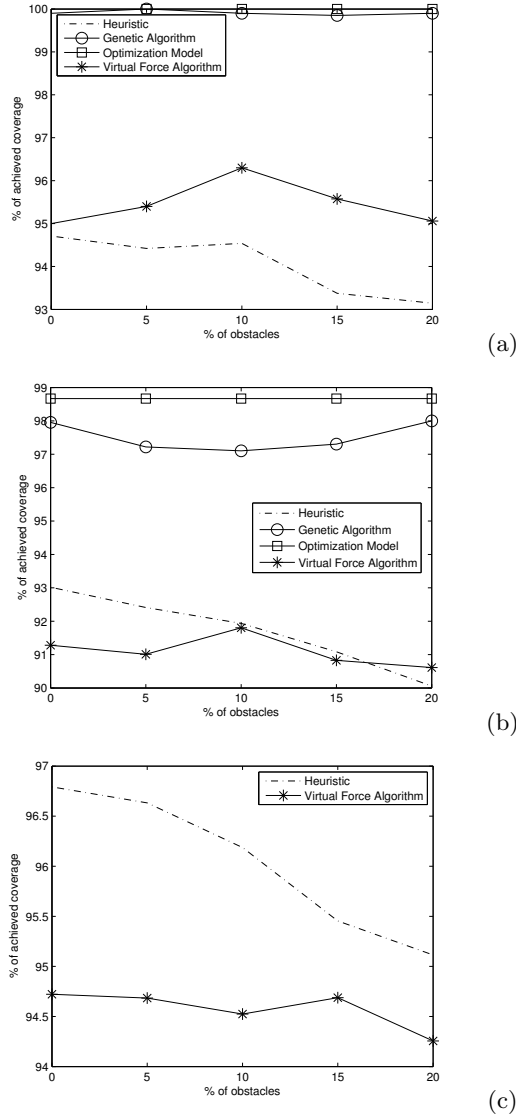


Fig. 5.6. Percentage of coverage when the grid size is 10x10 (a) 15x15 (b) and 25x25 (c) respectively and obstacles are considered .

associated to this more interesting zone. This second approach was followed with the Heuristic. In practice, we associated a certain “weight” to a specific zone/pixel that represents more interest to cover the zone without associating the specific number of sensor we need to cover this point/zone. Related to the

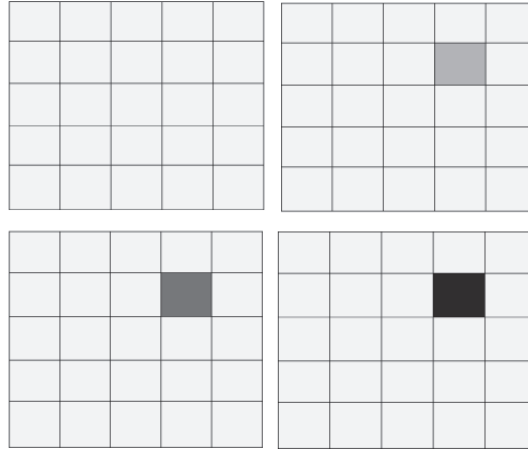


Fig. 5.7. An interesting zone (point) in the sensor field. The different degrees of gray correspond to different degrees of interest.

Zone of Interest (ZoIs) we introduced a modified version of the Basic-Heuristic considered, and we will refer to this as Ext-Heuristic (Extended-Heuristic). Specifically, in the Ext-Heuristic we take into account the information we have not only about the exterior frame as shown in Figure 5.1 but also the information regarding the interior frame. This extension is necessary when different “degrees” of interest are taken into account since an increased necessity to cover a certain zone could not be captured by only considering the second (exterior) frame. Specifically, we apply the same principle of the Basic-Heuristic and we calculate the total degree of coverage of a frame but nodes will move towards the zone with lesser coverage. In practice, let us consider the field as shown in Fig. 5.1 and focus on the gray node 1 inside the blue square, with the Basic-Heuristic we only take into account the more external frame (i.e., $[2, 2, 3, 2, 2], [1, 0, 1, 0, 0], [1, 1, 2, 2, 3], [1, 0, 0, 1, 1]$, etc. and we make the difference between the opposite sides (i.e., $[2, 2, 3, 2, 2]$ with $[1, 0, 1, 0, 0], [1, 1, 2, 2, 3]$ with $[1, 0, 0, 1, 1]$, etc). This difference give us the “direction” toward the current node has to move and the orientation will be determined by considering the lesser value of the weights (i.e. in the case of $[2, 2, 3, 2, 2]$ and $[1, 0, 1, 0, 0]$ the node will move toward the second). In our extended version we also take into account the contribution of the weight of the more internal frame by applying the same concept of the Basic-Heuristic (i.e. $[2, 3, 2]$, etc). The greater difference between the “opposite” frames will determine the direction of movement. It is worth to recall that where we do not explicit a different value of the sensing radius r we consider r equal to 2 as in this case.

In Fig. 5.8 we can observe as the Optimization Model is able to put exactly the number of nodes to “cover” the specific point and in Fig. 5.9 we observe as the coverage is kept.

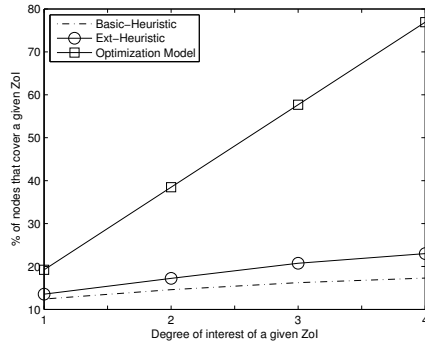


Fig. 5.8. The coverage of a specific ZoI (field size 10x10 and 26 nodes).

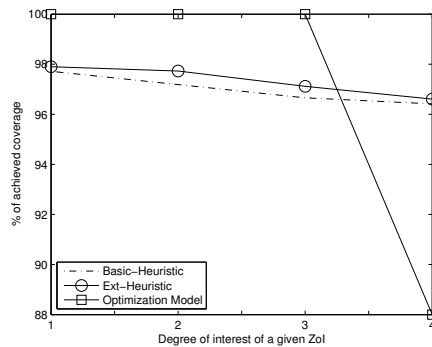


Fig. 5.9. The coverage of the whole field when a specific ZoI is defined (field size 10x10 and 26 nodes).

On the x – axis we put the “degree” of interest. In practice, we say that a specific zone of the sensor field needs to be highly monitored. It is worth to recall that the Optimization Model exactly defines the number of more nodes necessary to cover the ZoI that correspond to 5 nodes for 1 in Figs. 5.8 and 5.9, 10 nodes for 2, 15 nodes for 3 and 20 nodes for 4. This kind of definition of interest implies that a ZoI will be “covered” but in some other zone of the field there could be some holes as we can observe in Fig. 5.9. Both the Basic-Heuristic and the Ext-Heuristic move nodes in a soft way toward the ZoI by better keeping the general coverage of the field. It is worth to recall that both our techniques, the Basic-Heuristic and the Ext-Heuristic are only based on local information that could not be enough to answer to an increased request of nodes, but they are able to move some more nodes to the ZoI and simultaneously they keep the general coverage of the network. In practice, we observe that the number of nodes in the ZoI increase when the “degree” of requested nodes increases too, but not in an incisive manner. On the other

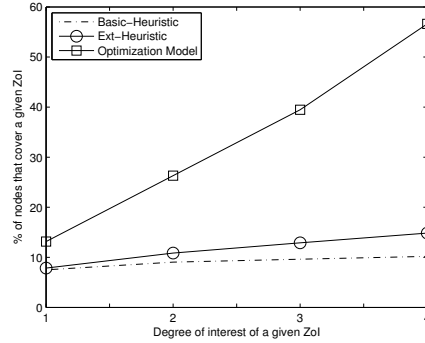


Fig. 5.10. The coverage of a specific ZoI (field size 15x15 and 38 nodes).

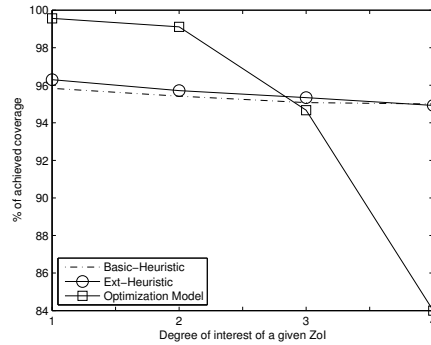


Fig. 5.11. The coverage of the whole field when a specific ZoI is defined (field size 15x15 and 38 nodes).

hand, we observe that the shape of the curve tend to be stable even if the zone is more “interesting”. this is due to two reasons: 1) as we already outlined the Basic-Heuristic is only based on local information and is not able to move far nodes towards more “interesting” zones; 2) the Basic-Heuristic tries to “respond” to the requests of the other zones and will try to arrange nodes to cover all the zones except the specific zone with obstacles. This means that we do not have all the nodes available for the ZoIs. A similar behavior is shown in Figs 5.10, 5.11, 5.12 and 5.13. In Fig. 5.10 we show the reactivity of our coverage techniques when the sensor field is 15X15 and the “degree” of interest increases as we already explained.

It is very interesting to notice as the greater is the sensor field, the worst the Optimization model is able to manage the coverage. On the other hand when the size of the field increases, both the heuristic, but above all the Ext-Heuristic, are able to arrange nodes in a way to smoothly “over-cover” the ZoI and at the same time to keep a good general coverage of the whole sensor

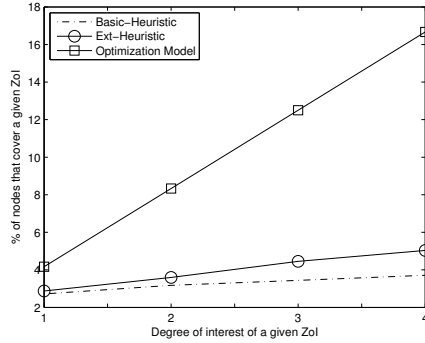


Fig. 5.12. The coverage of a specific ZoI (field size 25x25 and 120 nodes).

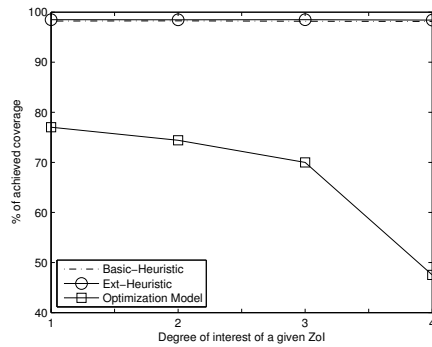


Fig. 5.13. The coverage of the whole field when a specific ZoI is defined (field size 25x25 and 120 nodes).

field. This behavior is really interesting since in real situation we can have the necessity to “over-cover” some specific zones for a limited time but we need to monitor the rest of the field in order to be able to move nodes in an appropriate fashion whether something interesting happens.

We considered another interesting kind of simulations based on the increasing of the sensing (coverage) radius r when a ZoI is defined in a sensor field of 15×15 , the total number of nodes is 38 and the “degree” of interest is 2. In Fig. 5.14 we considered the Basic-Heuristic-COV and the Ext-Heuristic-VIS. The terms COV and VIS are respectively for COVERAGE and VISIBILITY. In practice, we assume two kinds of radius, the first one is the coverage radius, that is the classical definition of radius, all the points that are inside a Lee sphere with radius r are covered from a sensor at the center of the Lee sphere. The visibility radius is the capability of a node to acquire the information (in this case the maps) at the distance r , but the coverage is always equal to 2. We retained useful the introduction of this different concept of radius in order

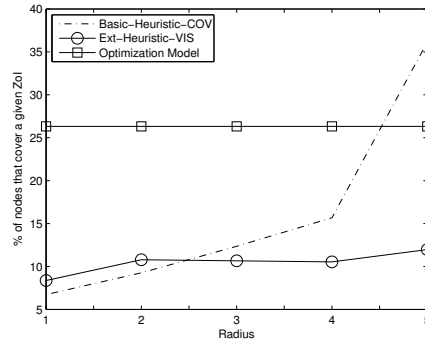


Fig. 5.14. The coverage of a specific ZoI with COV and VIS radius (field size 15x15, 38 nodes and “degree” of interest equal to 2).

to evaluate the effectiveness of more information in terms of maps and we considered that this kind of information can be easily obtained by considering a communication protocol where neighbor nodes change data (in a communication protocol we can easily figuring out that nodes can exchange maps till two-hops of distance). In Fig. 5.14 we show the behavior of the Optimization Model the Basic-Heuristic-COV with a variable coverage radius ranging from 1 to 5. To the contrary, the Ext-Heuristic-VIS considers the radius coverage r equal to 2, but it is able to obtain the maps until the variable distance ranging from 1 to 5. In Fig. 5.14 we can notice how only for the last point (where the coverage radius is equal to 5) there is an increasing of the coverage of the ZoI. On the other hand it is worth to outline that the same number of nodes cover more zones when the coverage radius is higher. In fact, the interesting aspect of these last Figures is in the observation of the Ext-Heuristic-VIS, where we only give our nodes more information by keeping the same coverage radius. As we can observe in Fig. 5.14 it seems that the more available information does not improve the concentration of the nodes in the ZoI.

On the other hand, the more information worsens the general coverage as we can observe in Fig. 5.15 where we show the coverage of the whole field with the conditions of the Fig. 5.14. In practice, our Ext-Heuristic behaves and reacts better when no too information are available since otherwise nodes will be pushed in many different direction simultaneously.

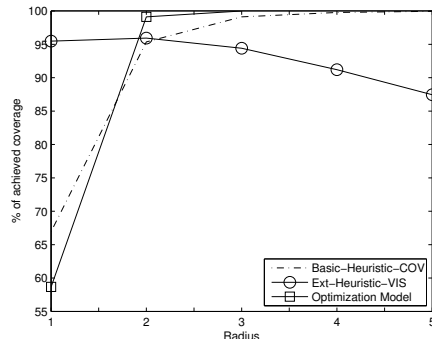


Fig. 5.15. The coverage of the whole field when specific ZoI is defined, with COV and VIS radius (field size 15x15, 38 nodes and “degree” of interest equal to 2).

5.7 Conclusion

In this chapter we proposed a new heuristic technique to move in an opportunistic fashion sensor devices in order to accomplish different coverage requests. This heuristic is extremely flexible, adaptive and works in a distributed fashion. In fact, based on local information it is able to “capture” the situation in the surrounding and nodes move in a greedy fashion by trying to cover the lesser covered zones. The surrounding information can both represent the information of how much a close point is covered and how much a node needs to be covered. This latter point allow to introduce the concept of “more interesting zone”, that is a zone that for many reasons needs to be highly monitored. The heuristic is extremely simple and can easily be implemented in simple sensor devices, because does not require much computation resources. In order to evaluate the effectiveness of this simple technique we considered three coverage approaches: an optimization model that give the optimal solution in terms of coverage, a genetic based approach that works in a centralized way and a virtual forces based technique. The different scenarios we considered show that our approach allows to reach results close to those of the optimization model and the genetic approach and in certain scenarios, when the size of the sensor field increases, our scheme is able to outperform the virtual based approach. Moreover, we introduced some obstacles in the sensor field in order to understand if our approach is able to adapt the movement of nodes in an effective way when obstacles are present. Results are encouraging also in this specific condition.

Conclusion

This PhD thesis has focused on communication networks composed of devices, whose tasks are no longer limited to data transmission. In fact, devices can be equipped with sensor for event detection, and several actuator also for allowing nodes to move.

The work done has shown as such devices need to be programmed in a new way through methodologies that permit the learning process and the emergence of behaviors that fit with the network issues. Methodologies useful for this purpose have been shown; some of them take inspiration from other disciplines, such as swarm intelligence and evolving neural networks, both inspired by biology.

Through this new way of "programming" the nodes, the network becomes a self-organized system where the components act in a cooperative way through simple locale rules learned from both the environment and the interactions with other nodes, and the global behaviour that emerges responds adequately to the dynamics of the surrounding environment that can also be particularly hostile.

Ringraziamenti

Sono molte le persone a cui voglio dire "Grazie" per avermi accompagnato con la loro presenza e il loro sostegno in questa esperienza di Dottorato di Ricerca.

Inizio con il ringraziare Enrico Natalizio, Valeria Loscrí e Pasquale Pace, che, dopo avermi seguito nel corso di laurea specialistica, mi hanno per primi trasmesso l'interesse e la passione per la ricerca, sostenendomi nell'intraprendere il percorso di Dottorato e Kate Price, per il supporto datomi con l'inglese oltre che motivazionale.

Ringrazio il mio supervisore, il Prof. Emanuele Viterbo e Gianluca Aloï per i loro insegnamenti e per la loro disponibilità nel guidarmi nel mio lavoro.

Ringrazio il Prof. Christian Bettstetter insieme al Mobile System Group del Lakeside Labs dell'Università di Klagenfurt, grazie a lui ho avuto la possibilità di vivere non solo un'esperienza lavorativa con i suoi tanti nuovi stimoli, ma un'esperienza di vita, che mi ha dato l'opportunità di misurarmi con me stesso.

Gli amici, in particolare Andrea Ferrise, Yongsoo Baek e Alireza Fasih incontrati durante le varie esperienze avute in questi tre anni di Dottorato, che, facendomi partecipe delle loro vite, mi hanno fatto riscoprire il valore dell'amicizia.

La mia fidanzata Francesca e la sua famiglia sempre presenti in tutti i momenti per me importanti.

Infine la mia famiglia che mi ha sempre sostenuto nelle mie decisioni con immenso affetto, credendo sempre in me.

Carmelo Costanzo

References

1. Modelling and Solving Optimal Placement problems in Wireless Sensor Networks, F. Guerriero, A. Violi, E. Natalizio, V. Loscrí, Costanzo C., in Elsevier Applied Mathematical Modelling.
2. Simulations of the impact of Controlled Mobility for Routing Protocol, V. Loscrí, E. Natalizio, C. Costanzo, in Eurasip journal on Wireless Communications and Networking, special issue Simulators and Experimental Testbeds Design and Development for Wireless Networks.
3. Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network, C. Costanzo, V. Loscrí, E. Natalizio, T. Razafindralambo, in Elsevier Computer Communication, Special issue on Wireless Sensor and Robot Networks: Algorithms and Experiments.
4. A Discrete Stochastic Process for Coverage Analysis of Autonomous UAV Networks, Evsen Yanmaz, Carmelo Costanzo, Christian Bettstetter, and Wilfried Elmenreich, accepted for International Workshop on Wireless Networking for Unmanned Aerial Vehicles (GLOBECOMM 2010), Miami, FL, USA, Dec 6, 2010.
5. D.K. Goldenberg, J. Lin, A.S. Morse, B.E. Rosen and Y.R. Yang, "Towards Mobility as a Network Control Primitive," in *Proceedings of ACM Mobihoc*, Page(s):163-174, 2004.
6. A.A. Somasundara, A. Kansal, D.D. Jea, D. Estrin and M.B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," in *IEEE Transactions on Mobile Computing*, Volume:5, Issue:8, Page(s):958-973, 2006.
7. R. Rao and G. Kesidis, "Purposeful Mobility for Relaying and Surveillance in Mobile Ad Hoc Sensor Networks," in *IEEE Transactions on Mobile Computing*, Volume:3, Issue:3, Page(s):225-232, 2004.
8. W. Wang, V. Srinivasan and K.C. Chua, "Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks," in *Proceedings of ACM MOBICOM*, Page(s):270-283, 2005.
9. J. Cortes, S. Martinez, T. Karatas and F. Bullo, "Coverage Control for Mobile Sensing Networks," in *IEEE Transactions on Robotics and Automation*, Volume:20, Issue:2, Page(s):243-255, 2004.
10. Z. Butler and D. Rus, "Controlling Mobile Sensors for Monitoring Events with Coverage Constraints," in *Proceedings of IEEE ICRA*, Volume:2, Page(s):1568-1573, 2004.

11. N. Bisnik, A. Abouzeid and V. Isler, "Stochastic Event Capture Using Mobile Sensors Subject to a Quality Metric," in *IEEE Transactions on Robotics*, Volume:23, Issue:4, Page(s):676-692, 2007.
12. Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," in *ACM Transactions on Embedded Computing Systems*, Volume:3, Page(s):61-91, 2004.
13. G. Wang, G. Cao and T.F. La Porta, "Movement-Assisted Sensor Deployment," in *IEEE Transactions on Mobile Computing*, Volume:5, Issue:6, Page(s):640-652, 2006.
14. W.K.G. Seah, K.Z. Liu, M.H. Ang Jr., J.G. Lim, and S.V. Rao, "TARANTULAS: Mobility-enhanced Wireless Sensor-Actuator Networks," in *Proceedings of IEEE Sensor Networks, Ubiquitous, and Trustworthy Computing*, Volume:1, Page(s):548-551, 2006.
15. J. Luo and J.P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," in *Proceeding of IEEE INFOCOM*, Volume:3, Page(s):1735-1746, 2005.
16. W. Zhao, M. Ammar and E. Zegura, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network," in *Proceedings of IEEE INFOCOM*, Volume:2, Page(s):1407-1418, 2005.
17. A.A. Somasundara, A. Ramamoorthy and M.B. Srivastava, "Mobile Element Scheduling with Dynamic Deadlines," in *IEEE Transactions on Mobile Computing*, Volume:6, Issue:4, Page(s):395-410, 2007.
18. A. Basu, B. Boshes, S. Mukherjee and S. Ramanathan, "Network Deformation: Traffic-Aware Algorithms for Dynamically Reducing End-to-end Delay in Multi-hop Wireless Networks," in *Proceedings of ACM MobiCom*, Page(s):100-113, 2004.
19. A. Kansal, M. Rahimi, D. Estrin, W.J. Kaiser, G.J. Pottie and M.B. Srivastava, "Controlled Mobility for Sustainable Wireless Sensor Networks," in *Proceedings of IEEE SECON*, Page(s):1-6, 2004.
20. Roh, H.-T. and Lee, J.-W. , Joint relay node placement and node scheduling in wireless networks with a relay node with controllable mobility. *Wireless Communications and Mobile Computing*, n/a. doi: 10.1002/wcm.1007
21. Ameer A. Abbasi, Mohamed Younis and Kemal Akkaya, Movement-Assisted Connectivity Restoration in Wireless Sensor and Actor Networks. In *IEEE Transactions on Parallel and Distributed Systems* archive Volume 20 Issue 9, September 2009 IEEE Press Piscataway, NJ, USA, doi:10.1109/TPDS.2008.246
22. Kemal Akkaya, Fatih Senel, Aravind Thimmapuram, Suleyman Uludag, "Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 258-271, Feb. 2010, doi:10.1109/TC.2009.120.
23. Abdel-Mageid, S. and Ramadan, R.A., Efficient deployment algorithms for mobile sensor networks, in *IEEE International Conference of Autonomous and Intelligent Systems (AIS)*.
24. E. Natalizio, V. Loscri, E. Viterbo, "Optimal Placement of Wireless Nodes for Maximizing Path Lifetime," in *IEEE Communication Letters*, Vol. 12, Issue 5 Page(s):362-364, 2008.
25. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," in *Elsevier Computer Networks*, Volume:38, Issue:2, Page(s):393-422, 2002.

26. Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulaz, and Eric Bonabeau, "Self-Organization in Biological Systems", Princeton University Press, 2001.
27. Falko Dressler, "Self-Organization in Ad-Hoc Networks: Overview and Classification, University of Erlangen, Dept. of Computer Science 7, Technical Report 02/06.
28. Collier, Travis C. and Taylor, Charles, "Self-organization in sensor networks", in *Journal Parallel Distrib. Comput.*, vol. 64, issue 7, pages 866-873, July 2004.
29. M. Bhatt, R. Chokshi, S. Desai, S. Panichpapiboon, N. Wisitpongphan, and O. K. Tonguz, Impact of mobility on the performance of ad hoc wireless networks, in *IEEE Vehicular Tech. Conf.(VTC-Fall03)*, vol. 5, Orlando USA, 2003, pp. 30253029.
30. W. Elmenreich, R. D'Souza, C. Bettstetter, H. De Meer, "A Survey of Models and Design Methods for Self-Organizing Networked Systems", *Proceedings of the 4th International Workshop on Self-Organizing Systems*, 2009.
31. Olariu, S. and Stojmenovic, I. 2006. Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting. In *25th IEEE Conference on Computer Communications (IEEE INFOCOM 2006)*. Barcelona, Spain.
32. Giridhar, A. and Kumar, P. 2005. Maximizing the Functional Lifetime of Sensor Networks. In *4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)*. Los Angeles, CA.
33. I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks", *ACM Trans. Sen. Netw.*, vol. 5, pages 5:1-5:39, February 2009.
34. Rajeev Shorey, A. Ananda, Mun Choon Chan, Wei Tsang Ooi, "Mobile, wireless, and sensor networks: technology, applications, and future directions", *IEEE Press*, 2006.
35. D. W. Gage, "Command control for many-robot systems", *Proc. 19th Annual AUVS Technical Symp.* Reprinted in *Unmanned Syst. Mag.* 10(4):2834 (Jan. 1992).
36. J. ORourke, *Art Gallery Theorems and Algorithms*, Oxford Univ. Press, Oxford, UK, 1987.
37. P. Gupta and P. R. Kumar, "The capacity of wireless networks", *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388-404, 2000.
38. Markus Quaritsch, Emil Stojanovski, Christian Bettstetter, Gerhard Friedrich, Hermann Hellwagner, Bernhard Rinner, Michael Hofbaur, Mubarak Shah, "Collaborative Microdrones: Applications and Research Challenges" in *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, pages 38:1-38:7, 2008.
39. Alberto Sanfeliu, Norihiro Hagita and Alessandro Saffiotti, "Network robot systems", in *Robotics and Autonomous Systems Journal*, vol. 56, number 10, pages 793-797, 2008.
40. I. F. Akyildiz, I. H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges", *Ad Hoc Networks Journal (Elsevier)*, Vol. 2, No. 4, pp. 351-367, October 2004.
41. Al-Karaki, J.N. and Kamal, A.E.: *Routing Techniques in Wireless Sensor Networks: A Survey*. *IEEE Personal Communications*, Vol.11, Issue 6, Page(s):6-28, 2004.

42. E. Natalizio and V. Loscri, "Controlled Mobility in Mobile Sensor Networks: Advantages, Issues and Challenges", in Springer Telecommunication Systems, Special Issue on Recent Advance in Mobile Sensor Networks.
43. E. Natalizio, V. Loscri, F. Guerriero, A. Violi, Energy spaced placement for bidirectional data flows in wireless sensor network, *IEEE Commun. Lett.* 13-1 (2009) 22 - 24.
44. E. Natalizio, V. Loscri, C. Costanzo, F. Guerriero, A. Violi, Optimization Models for Determining Performance Benchmarks in Wireless Sensor Networks, *Proc. Third International Conference on Sensor Technologies and Applications* (2009) 333-338.
45. Y.-C. Wang, C.-C. Hu, Y.-C. Tseng, Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network, *IEEE Trans. Mob. Comput.* 7-2 (2008) 262-274.
46. P. Cheng, C.-N. Chuah, X. Liu, Energy-aware node placement in wireless sensor networks, *Proc. IEEE Glob. Telecommun. Conf.* 5 (2004) 3210-3214.
47. S. Kim, J.-G. Ko, J. Yoon, H. Lee, Multiple-Objective Metric for Placing Multiple Base Stations in Wireless Sensor Networks, *Proc. 2nd Internat. Symp. on Wirel. Pervasive Comput.* (2007) 627-631.
48. S. Habib, M. Safar, Sensitivity Study of Sensors' Coverage within Wireless Sensor Networks, in *Proc. of 16th Internat. Conf. on Comput. Commun. and Netw.* (2007) 876-881.
49. A. Krause, C. Guestrin, A. Gupta, J. Kleinberg, Near-optimal sensor placements: maximizing information while minimizing communication cost, *Proc. 5th Internat. Conf. Inf. Process. in Sens. Netw.* (2006) 2-10.
50. D.J. Chmielewski, T. Palmer, V. Manousiouthakis, On the theory of optimal sensor placement, *AICHE journal* 48-5 (2002) 1001-1012.
51. Y. Chen, C.-N. Chuah, Q. Zhao, Sensor placement for maximizing lifetime per unit cost in wireless sensor networks, *MILCOM 2* (2005) 1097-1102.
52. I. Dietrich, F. Dressler, On the lifetime of wireless sensor networks, technical report Erlangen-Nurnberg: Friedrich-Alexander-Universitaet (2006).
53. R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, D. Estrin, Habitat monitoring with sensor networks, *Commun. of the ACM* 47-6 (2004) 3440.
54. L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, J. Porter, Luster: wireless sensor network for environmental research, *ACM Proc. 5th Internat. Conf. on Embed. Netw. Sens. Syst.* (2007) 103116.
55. T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, S. Brosnan, Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network, *Proc. 2nd IEEE Workshop on Pract. Issues in Build. Sens. Netw. Appl.* (2007) 799806.
56. W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in *MobiHoc 04: Proc. 5th ACM Internat. Symp. on Mobile Adhoc Netw. and Comput.* (2004) 187198.
57. C. Frank, K. Romer, Algorithms for generic role assignment in wireless sensor networks, *Proc. 3rd Internat. Conf. on Embed. Netw. Sens. Syst.* (2005) 230242.
58. A. Ledeczi, P. Volgyesi, M. Maroti, G. Simon, G. Balogh, A. Nadas, B. Kusy, S. Dora, G. Pap, Multiple simultaneous acoustic source localization in urban terrain, *Proc. 4th Internat. Symp. Inform. Process. in Sens. Netw.*, (2005) 69.

59. R. Chellappa, G. Qian, Q. Zheng, Vehicle detection and tracking using acoustic and video sensors, Proc. Internat. Conf. on Acoustics, Speech and Signal Process., (2004) 793796.
60. T. Bokarev, W. Hu, S. Kanhere, B. Ristic, N. Gordon, T. Bessell, M. Rutten, S. Jha, Wireless sensor networks for battlefield surveillance, Proc. Land Warfare Conf., (2006).
61. W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. on Wirel. Commun. 1-4 (2002) 660-670.
62. J. Wu, Handbook on theoretical and algorithmic aspects of sensor ad hoc wireless, and peer-to-peer networks, Auerbach Publication, US.
63. B. Carbunar, A. Grama, J. Vitek, O. Carbunar, Redundancy and coverage detection in sensor networks, ACM Transaction on Sensor Networks 2 (2006) 94–128.
64. H. Gupta, Z. H. Zhou, S. R. Das, Q. Gu, Connected sensor cover: Self-organization of sensor networks for efficient query execution, IEEE/ACM Transaction on Networks 14 (2006) 55–67.
65. S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, K. Zhang, Deploying wireless sensor networks under limited mobility constraints, IEEE Transactions on Mobile Computing 6(10) (2007) 1142–1157.
66. Y. C. Wang, C. C. Hu, Y. C. Tseng, Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks, Proceedings of the First International Conference on Wireless Internet (2005) 114-121.
67. M. A. Batalin, G. S. Sukhatma, The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment, IEEE Transaction Robotics 23 (2007) 661-675.
68. X. Bai, D. Xuan, Z. Yun, T. H. Lai, W. Jia, Complete optimal deployment patterns for full-coverage and k -connectivity ($k \leq 6$) wireless sensor networks, in: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '08, 2008, pp. 401–410.
69. C. Wu, Y. Lee, K.C. and Chung, A delaunay triangulation based method for wireless sensor network deployment, Computer Communications 30 (2007).
70. S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, in: INFOCOM 2001. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2001, pp. 1380–1387.
71. J. Chen, E. Shen, Y. Sun, The deployment algorithms in wireless sensor networks: A survey, Information Technology Journal 8 (3) (2009) 293–301.
72. B. Wang, H. B. Lim, D. Ma, A survey of movement strategies for improving network coverage in wireless sensor networks, Computer Communications 32 (13-14) (2009) 1427– 436.
73. N. A. A. Aziz, K. A. Aziz, W. Z. W. Ismail, Coverage strategies for wireless sensor networks, World Academy of Science, Engineering and Technology 50 2009.
74. A. Howard, S. Poduri, Potential field methods for mobile-sensor-network deployment, IEEE Transactions on Robotics 23 (4) (2007) 661-675.
75. X. Shen, J. Chen, Z. Wang, Y. Sun, Grid scan: A simple and effective approach for coverage issue in wireless sensor networks, in: IEEE International Communications Conference, Vol. 8, 2006, pp. 3480-3484.

76. Y. C. Wang, C. C. Hu, Y. C. Tseng, Efficient placement and dispatch of sensors in a wireless sensor network, *IEEE Transactions on Mobile Computing* 7 (2) (2008) 262-274.
77. G. Wang, G. Cao, T. Porta, Movement-assisted sensor deployment, in: In IEEE INFOCOM'04, Vol. 4, 2004, pp. 2469-2479.
78. S. Megerian, F. Koushanfar, M. Potkonjak, M. Srivastava, Worst and best-case coverage in sensor networks, *IEEE Transaction on Mobile Computing* 4 (2005) 84-92.
79. C.-Y. Chang, C.-T. Chang, Y.-C. Chen, H.-R. Chang Obstacle-resistant deployment algorithms for wireless sensor networks, *IEEE Transaction on Vehicular Technology* 58 (6) (2009) 2925-2941.
80. N. Bartolini, T. Calamoneri, E. G. Fusco, A. Massini, S. Silvestri, Push&pull: autonomous deployment of mobile sensors for a complete coverage, *Wireless Networks* 16 (2010) 607-625.
81. N. Heo, P. Varshney, A distributed self spreading algorithm for mobile wireless sensor networks, in: *IEEE Wireless Communications and Networking Conference*, Vol. 3, 2003, pp. 1597-1602.
82. F. Arvin, K. Samsudin, A. R. Ramli, A short-range infrared communication for swarm mobile robots, in: *Proceedings of the 2009 International Conference on Signal Processing Systems*, IEEE Computer Society, 2009, pp. 454-458.
83. J. Majchrzak, M. Michalski, G. Wiczynski, Distance estimation with a long-range ultrasonic sensor system, *IEEE Sensors Journal* 9 (2009) 767-773.
84. H. Zhang, L. Wang, R. Jia, J. Li, A distance measuring method using visual image processing, in: *2nd International Congress on Image and Signal Processing, CISP '09*, 2009, pp. 1-5.
85. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, G. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: *The 1st International Conference on Embedded Networked Sensor System*, 2003, pp. 28-39.
86. A. Laouiti, A. Qayyum, L. Viennot, Multipoint relaying: an efficient technique for flooding in mobile wireless networks, in: *35th Annual Hawaii International Conference on System Sciences HICSS*, 2002, pp. 3898-3907.
87. H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," Technical Report UIUC, UIUCDCS-R-2003-2351, 2003.
88. A. Howard, M.J. Mataric, and G.S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," in *Autonomous Robots, special issue on intelligent embedded systems*, Sept. 2002.
89. A. Howard, M. Mataric and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, 2002.
90. S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE Intl. Conf. on Robotics and Automation (ICRA04)*, 165-171, 2004.
91. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," in *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660-670, October 2002.
92. C. Costanzo and V. Loscri and E. Natalizio, "Distributed Virtual-Movement Scheme for Improving Energy Efficiency in Wireless Sensor Networks", *Proceedings of ACM MSWIM*, 2009, pages 297-304.

93. B. Shafai and J. Chen and M. Kothandaraman, "Explicit Formulas for Stability Radii of Nonnegative and Metzlerian Matrices", *IEEE Transactions on Automatic Control*, February 1997, vol. 42, number 2, pages 265-270.
94. J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks", *Proceedings of IEEE INFOCOM*, 2000, vol. 1, pages 22-31.
95. D. P. Bertsekas, "Nonlinear Programming", Athena Scientific, 1995.
96. M. Marcus and H. Minc, "A Survey of Matrix Theory and matrix inequalities, Dover Publications, 1969.
97. R. Horn and C. R. Johnson, "Matrix Analysis, Cambridge University Press, 1995.
98. Kling, R.M. "Intel Motes: advanced sensor network platforms and applications." In *Digest of IEEE MTT-S International Microwave Symposium*, 2005.
99. Mahoney, R.M. "Robotic Products for Rehabilitation: Status and Strategy." In *Proceedings of the International Conference on Rehabilitation Robotics ICORR*, the Bath Institute of Medical Engineering, Bath University, UK, 1997.
100. Topping, M., Hegaty, J. R. "The Potential of Low-Cost Computerised Robot Arms as Aids to Independence for People with Physical Disability." In *Proceedings of the 1st International Workshop Domestic Robots and the 2nd Workshop on Medical and Healthcare Robotics*, Newcastle upon Tyne, UK, 1989.
101. S. Singh, M. Woo and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," in *Proceedings of the 4th Annual IEEE/ACM International Conference on Mobile Computer and Network, MOBICOM*, pp. 181-90, Dallas, October 1998.
102. L. Jun and J.P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM 2005*, pp. 1735-1746, vol.3, Miami, March 2005.
103. I.D. Chakeres and E.M. Belding-Royer. "AODV Routing Protocol Implementation Design" in *Proceedings of the International Workshop on Wireless Ad Hoc Networking (WWAN)*, Tokyo, Japan, March 2004.
104. C.E. Perkins, E.M. Belding-Royer, and I. Chakeres. "Ad Hoc On Demand Distance Vector (AODV) Routing." in *IETF Internet draft, draft-perkins-manet-aodvbis-00.txt*, Oct 2003.
105. V. D. Park and M. S. Corson, A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, in *Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM 1997*, Apr. 1997.
106. E. Natalizio, V. Loscrí and E. Viterbo, "Optimal Placement of Wireless Nodes for Maximizing Path Lifetime," in *IEEE Communications Letters*, vol. 12, no. 5, pp. 362-364, May 2008.
107. E. Natalizio, V. Loscrí, F. Guerriero, A. Violi, "Energy Spaced Placement for Bidirectional Data Flows in Wireless Sensor Network," in *IEEE Communications Letters*, vol. 13, no. 1, pp. 22-24, January 2009.
108. V. Loscrí, E. Natalizio, C. Costanzo, F. Guerriero and A. Violi, "Optimization Models for Determining Performance Benchmarks in Wireless Sensor Networks," in *Proceedings of Third International Conference on Sensor Technologies and Applications, SENSORCOMM 2009*, Athens, Greece, June 2009.
109. <http://www.isi.edu/nsnam/ns/>
110. M. Grossglauser, and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *IEEE/ACM Transactions on Networking*, 10(4), pp. 477-486.

111. A.E. Gamal, J.Mammen, B. Prabhakar and D. Shah, "Throughput-delay trade-off in wireless networks," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM 2004*, Hong-Khong, pp. 464-475, March 2004.
112. N. Bansal and Z. Liu, "Capacity, Delay and Mobility in Wireless ad-hoc Networks," in *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM 2003*, San-Francisco, California, pp. 1553-1563, March-April 2003.
113. R.M. Moraes, H.R. Sadjadpour, and J.J. Garcia-Luna-Aceves, "Mobility-capacity-delay trade-off in wireless ad hoc networks," in *Elsevier Ad Hoc Networks*, 4(5), pp. 607-620, 2006.
114. E.Lee, S. Park, D. Lee, Y. Choi, F. Yu, and S.H. Kim, "A Predictable Mobility-based Communication Paradigm for Wireless Sensor Networks," in *Proceedings of Asia-Pacific Conference on Communications*, pp. 373-376, 2007.
115. Z. Zhi, D. Guanzhong, L. Lixin and Z. Yuting, "Relay-based routing protocols for space networks with predictable mobility," in *Proceedings of SPIE*, 2005.
116. S. Merugu, M. Ammar, and E. Zegura, "Routing in space and time in networks with predictable mobility" in *Technical Report GIT-CC No. GIT-CC-04-07*.
117. A. Howard, M. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential field: a distributed scalable solution to the area coverage problem," in *Proceedings of DARS*, 2002.
118. D. Wang, J. Liu, and Q. Zhang, "Probabilistic Field Coverage using a Hybrid Network of Static and Mobile Sensors," in *Proceedings of the International Workshop on Quality of Service*, Chicago, IL, USA, June 20-22, 2007.
119. L. Schwiebert, S.K.S. Gupta, and J. Weinmann, "Research Challenges in Wireless Networks of Biomedical Sensors," *Proc. MobiCom 2001*, pp. 151-165, 2001.
120. Bokareva, T., Hu, W., Kanhere, S., Ristic, B., Gordon, N., Bessel, T., Rutten, M., Jha, S.: *Wireless Sensor Networks for Battlefield Surveillance*. In: *Proceedings of the Land Warfare Conference (LWC 06)*, Brisbane, Australia, October 2006.
121. Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler and John Anderson, "Analysis of Wireless Sensor Networks for Habitat Monitoring," in *Wireless Sensor Networks 2004*, Springerlink, VI, 399-423, DOI: 10.1007/1-4020-7884-6-18.
122. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, *Wireless Sensor Networks for Habitat Monitoring*, *Proc. First ACM Intl Workshop Wireless Sensor Networks and Applications (WSNA 02)*, Sept. 2002.
123. K.Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks", in *IEEE Transactions on Computers*, Vol.51, No. 12, December 2002.
124. P. Ray, P. Varshney, "Estimation of spatially distributed processes in wireless sensor networks with random packet loss," *IEEE Trans. Wireless Communications* 8 (6) (2009) 3162-3171.
125. K. K. Rachuri, C. Murthy, "Energy efficient and scalable search in dense wireless sensor networks," in *IEEE Trans. Computers* 58 (6) (2009) 812-826.
126. Z. Zhou, S. Das, H. Gupta, "Connected k-coverage problem in sensor networks," in *Proceedings of ICCCN*, Chicago, IL, 2004.
127. C. F. Huang, Y. C. Tseng, H. L. Wu, "Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network," in *ACM Trans. Sensor Networks* 3 (1) (2007) 5.

128. N. Heo, P. K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in: Proceedings of IEEE SMC, Washington, D.C., 2003.
129. Y. Zou, K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," IEEE Trans. Computers 54 (8) (2005) 978-991.
130. W. C. Ke, B. H. Liu, M. J. Tsai, "Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete," IEEE Trans. Computers 56 (5) (2007) 710-715.
131. W.C. Ke, B. H. Liu and M. J. Tsai, "The CRITICAL-SQUARE-GRID COVERAGE Problem in Wireless Sensor Networks is NP-Complete" in Computer Networks 55(9): 2209-2220 (2011).
132. S. Golomb and L. Welch, "Perfect codes in the Lee metric and the packing of polyominoes," in SIAM J. Appl Math., 18:302-317, 1970.
133. A. W. Paeth, *Distance Approximations and Bounding Polyhedra*. In A. W. Paeth, editor, *Graphics Gems V*, Chapter II, Pages 78–87, USA: Academic Press, 1995.
134. D. Bertsimas, J. N. Tsitsiklis *Introduction to Linear Optimization*, USA: Athena Scientific, 1997.
135. F. Hoffmann, M. Kaufmann, K. Kriegel, "The art gallery theorem for polygons with holes," in: Proceedings of IEEE FOCS, New York, NY, 1991.
136. J. B. M. Melissen, P. C. Schuur, "Covering a rectangle with six and seven circles," in Discrete Applied Math. 99 (1) (2000) 149-156.
137. K. J. Nurmela, "Conjecturally optimal coverings of an equilateral triangle with up to 36 equal circles," in Experimental Math. 9 (2) (2000) 241-250.
138. K. J. Nurmela, P. R. J. Ostergard, "Covering a square with up to 30 equal circles," in Tech. rep., HUT-TCS-A62 (2000).
139. M. Ahmadi and P. Stone, "A multi-robot system for continuous area sweeping tasks", Proceedings of International Conference on Robotics and Automation (ICRA'06), pp. 1724-1729, 2006.
140. L. E. Parker, "Distributed algorithms for multi-robot observation of multiple moving targets", Autonomous Robots, vol. 12, pp. 231-255, 2002.
141. L. Gray, "A Mathematician Looks at Wolfram's New Kind of Science", Notices of the AMS, vol. 50, pp. 200-211, Feb. 2003.
142. Kuiper, E. and Nadjm-Tehrani, S., "Mobility Models for UAV Group Reconnaissance Applications", Proc. Intl. Conf. Wireless and Mobile Communications (ICWMC), 2006.
143. Elfes, A., "Using Occupancy Grids for Mobile Robot Perception and Navigation", Computer, vol. 22, pp. 46-57, June 1990.
144. L. Kleinrock, *Queueing Systems Volume I: Theory*, Wiley, 2005.
145. I. Fehérvári and W. Elmenreich, "Evolutionary Methods in Self-Organizing System Design", Proceedings of the 2009 International Conference on Genetic and Evolutionary, pp. 10-15, Las Vegas, USA, 2009.
146. I. Fehervari and W. Elmenreich, "Evolving Neural Network Controllers for a Team of Self-organizing Robots", Journal of Robotics, 2010.
147. Engelhart, D.C. and Sivasubramaniam, A. and Barrett, C.L. and Marathe, M.V. and Smith, J.P. and Morin, M., "A spatial analysis of mobility models: application to wireless ad hoc network simulation", Proc. Simulation Symp., pp. 3-42, 2004.

148. Leonard, J.J. and Durrant-Whyte, H.F., "Simultaneous map building and localization for an autonomous mobile robot", Proc. IEEE/RSJ Intl. Workshop on Intelligent Robots and Systems (IROS), vol. 3, pp. 1442-1447, November 1991.
149. Lin, G. and Noubir G and Rajmohan Rajaraman, "Mobility models for ad hoc network simulation", Proc. IEEE Conf. Comp. (INFOCOM), vol. 1, pp. 454-463, 2004.
150. B. Liu and P. Brass and O. Dousse and P. Nain and D. Towsley, "Mobility Improves Coverage of Sensor Networks", Proc. ACM Intl. Symp. Mob. Ad hoc Net. Comp. (MobiHoc), pp. 300-308, 2005.
151. Yanmaz, E. and Bettstetter, C., "Area Coverage with Unmanned Vehicles: A Belief-Based Approach", Proc. IEEE Vehicular Technology Conf. (VTC), pp. 1-5, 2010.
152. Elmenreich, W. and Klingler, G., "Genetic evolution of a neural network for the autonomous control of a fourwheeled robot", In Sixth Mexican International Conference on Artificial Intelligence (MICAI'07), November 2007.
153. N.A.B.A. Aziz and A. W. Mohemmed and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram", International Conference on Networking, Sensing and Control, ICNSC '09, March 2009, pages 602-607.
154. T. Blackwell, J. Branke, and X. Li, "Particle Swarms for Dynamic Optimization Problems", in Swarm Intelligence Natural Computing Series, 2008, Part II, 193-217, DOI:10.1007/978-3-540-74089-6_6.
155. E. Bonabeau, M. Dorigo and G. Theraulaz "Swarm Intelligence: From Natural to Artificial Systems" Oxford University Press, 1999. ISBN 0-19-513159-2.
156. C. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities and challenges," in Proceedings of the IEEE, Vol. 91, No. 8. (August 2003), pp. 1247-1256, DOI:10.1109/JPROC.2003.814918.
157. C. Costanzo, V. Loscri', E. Natalizio, T. Razafindralambo, "Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network," Elsevier Computer Communications, Special Issue on Wireless Sensor and Robot Networks: Algorithms and Experiments, DOI: <http://dx.doi.org/10.1016/j.comcom.2011.09.004>.
158. F. Kribi, P. Minet, A. Laouiti, "Redeploying mobile wireless sensor networks with virtual forces," Wireless Days (WD), 2009 2nd IFIP , pp.1-6, 15-17 Dec. 2009, DOI: 10.1109/WD.2009.5449665.
159. R.V. Kulkarni, G.K. Venayagamoorthy, "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol.41, no.2, pp.262-267, March 2011, DOI: 10.1109/TSMCC.2010.2054080.
160. S. Li, C. Xu, W. Pan and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets," Information Fusion, 2005 8th International Conference on , vol.2, pp. 7, 25-28 July 2005 DOI: 10.1109/ICIF.2005.1592051.
161. T. Li, Y. ChongChong, Y. Minghua, "A multi-center self-deployment algorithm of mobile sensor network," Computer Science and Education (ICCSE), 2010 5th International Conference on, pp.965-968, 24-27 Aug. 2010, DOI: 10.1109/ICCSE.2010.5593452.
162. R. Olfati-Saber, J. A. Fax and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in Proceedings of the IEEE, 2007.
163. P. A. Tipler, "Physics for Scientists and Engineers", in Worth Publishers, 1991.

164. X. Wang, S. Wang, and J. J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, pp. 354-370, 2007.
165. Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization Based on Virtual Forces, " in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003*, pp. 1293-1303, vol.2, 30 March-3 April, 2003.