



UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica

Dottorato di Ricerca in

Ingegneria dei Sistemi e Informatica

CICLO

XXVII

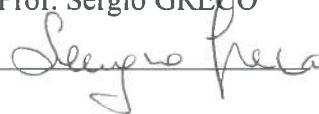
TITOLO TESI

**Design and Performance Evaluation
of Algorithms for Wireless
Self-Organizing Systems**

Settore Scientifico Disciplinare

ING-INF/03

Coordinatore: Ch.mo Prof. Sergio GRECO

Firma 


Supervisore/Tutor: Ing. Valeria LOSCRI

Firma 

Ing. Gianluca ALOI

Firma 

Dottorando: Dott. Rosario SURACE

Firma 

Rosario Surace

Design and Performance
Evaluation of Algorithms for
Wireless Self-Organizing Systems

1 Dicembre 2014

Contents

Abstract (English version)	1
Abstract (Italian version)	3
1 Self-Organizing Systems	5
1.1 Wireless Sensor Networks	7
1.2 Robot Networks	9
1.3 UAV Networks	10
2 Impact of the Propagation Environment on CMA	13
2.1 Related Works	14
2.2 Propagation and Energy Models	15
2.2.1 Propagation Models	15
2.2.2 Energy Models	16
2.3 Controlled Mobility Algorithms	17
2.4 Results	18
2.4.1 Results of the free-space model	20
2.4.2 Results of the two-ray ground model	21
2.4.3 Results of the generic model	22
2.5 Conclusions	24
3 Algorithms to Film Sport Events with Flying Robots	25
3.1 Related Works	26
3.2 Definitions and initial assumptions	27
3.2.1 Action	27
3.2.2 Event	29
3.2.3 Viewer satisfaction	29
3.2.4 Relevant assumptions related to the drones	29
3.3 Distributed algorithms for dynamic VRP-STW	30
3.3.1 Nearest Neighbor	30
3.3.2 Nearest Neighbor-Division Field	30

3.3.3	Nearest Neighbor with Specular Repositioning	31
3.3.4	Nearest Neighbor with Quasi-Specular Repositioning	31
3.3.5	Ball Movement Interception	31
3.4	Performance Results	32
3.4.1	Performance evaluation varying detour factor	33
3.4.2	Comparison of Positioning Techniques	35
3.5	A Multi-objective Approach for UAV Problem	37
3.5.1	Problem Statement and Mathematical Formulation	38
3.5.2	The solution approaches	46
3.5.3	Computational Experiments	47
3.6	Conclusion	54
4	Multi-Objective Evolving Neural Network	55
4.1	Related Works	57
4.2	Smart mobile devices supporting SDR	58
4.3	Neural Network and Genetic Algorithm	60
4.3.1	The Neural Network	61
4.3.2	The Genetic Algorithm	62
4.3.3	Connectivities and Communication Complexity	63
4.3.4	BER Computation	64
4.3.5	Transmitted Energy Computation	64
4.3.6	Cut-off rate curves	65
4.4	The optimization model	67
4.5	Validations, Simulations and Results	70
4.5.1	Validation of the Optimizazion Model	71
4.5.2	Fixed nodes analysis supporting SDR	72
4.5.3	Mobile nodes analysis supporting SDR	72
4.5.4	Mobile nodes analysis with variable SDR nodes	74
4.5.5	Varying the percentange of mobile nodes	75
4.6	Video Surveillance Applications with Ultra-Low Power Sensor	77
4.6.1	Reference Model and Problem Formulation	79
4.6.2	Evolutionary algorithm for neural network training	81
4.6.3	Performance Evaluation	82
4.7	Conclusion	84
5	Decentralized Time-Synchronized Channel Swapping	85
5.1	Related Work	86
5.1.1	Novelty, Contributions and work Organization	89
5.2	PCO-based SYNC/DESYNC for DT-SCS	90
5.2.1	Introduction to the Basic Concept	90
5.2.2	Coupling via SYNC and DESYNC	91
5.2.3	Protocol Description	94
5.2.4	Discussion	98
5.3	Protocol Analysis	99
5.3.1	Balancing and Stability	99

5.3.2	Connectivity	101
5.3.3	Estimation of Energy Consumption	103
5.4	Simulation Results	104
5.4.1	Node Balancing and Connectivity	104
5.4.2	Convergence Time of DT-SCS versus TSCH	106
5.4.3	Bandwidth Efficiency	109
5.5	Experiments With TelosB Motes	110
5.5.1	Energy Consumption	111
5.5.2	Results Under Interference	111
5.5.3	Bandwidth Results	113
5.6	Conclusion	114
Conclusion		115
References		117
Reference of the Candidate		127

Abstract (English version)

The work done during the PhD course involves the study of the Self-Organization of wireless sensors, robots and UAV networks. In particular, this thesis investigates how each node composing the system can take advantage from the *Self-Organization* and from *mobility*, in a way to optimize some networks parameters as coverage and energy consumption.

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 [1].

Mobility, although still for some types of systems is not considered a primitive of the network: in recent years has been the subject of many studies just as useful feature to achieve certain objectives, not least the energy consumption in transmission.

The network issues has been addressed using different approaches from the theoretical studies aimed at finding the maximum achievable performance benchmarks, through the introduction of appropriate optimization models, the proposal of distributed heuristics and more realistic communication protocols, and the use of biology-inspired mechanisms, such as genetic algorithms (GA) 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 hard scenarios (*i.e.* environment disasters).

The rest of the thesis is organized as follows: in Chapter 1 background on Self-Organizing Systems is given. In Chapter 2 we investigate on the impact of the Propagation Environment on Controlled Mobility Algorithms; distributed heuristics to Film Sport Events with Flying Robots in Chapter 3 and Bio-Inspired approaches in Chapter 4. Finally, a new communications protocol for WSN called Decentralized Time-Synchronized Channel Swapping is analyzed in Chapter 5.

Abstract (Italian version)

Il lavoro svolto durante il corso di dottorato di ricerca ha previsto lo studio dell'auto-organizzazione, meglio conosciuta come *Self-Organization*, in reti wireless di sensori, robot e UAV. In particolare, questa tesi indaga come ogni nodo che compone il sistema può trarre vantaggio dalla *Self-Organization* e dalla *mobilità*, in modo da ottimizzare alcuni parametri di rete come la copertura ed il consumo energetico.

La *Self-Organization* è un processo, il cui schema a livello globale, emerge unicamente dalle numerose interazioni tra i componenti di livello inferiore del sistema. Le regole che specificano le interazioni tra i componenti sono eseguite utilizzando solo informazioni locali, senza riferimento al modello globale [1].

La mobilità, anche se ancora per alcune tipologie di sistemi non è considerata una primitiva di rete, negli ultimi anni è stata oggetto di molti studi proprio come caratteristica utile al raggiungimento di determinati obiettivi, non ultimo quello del consumo energetico in trasmissione.

I problemi di rete sono stati affrontati usando approcci diversi, a partire dagli studi teorici, volti a trovare il rendimento massimo ottenibile (benchmarks) attraverso l'introduzione di adeguati modelli di ottimizzazione, passando per la proposta di euristiche distribuite e protocolli di comunicazione più realistici, oltre che attraverso l'uso di meccanismi bio-ispirati, come gli algoritmi genetici (GA) e le reti neurali (NN). Lo scopo di questo tipo di approccio è quello di muoversi nella direzione di reti che sono in grado di auto-organizzarsi adattandosi alle diverse condizioni ambientali ed a scenari dinamici nonché difficili (*es.* disastri ambientali).

Il resto della tesi è organizzata come segue: nel Capitolo 1 verrà fornito un background sui Self-Organizing Systems. Nel Capitolo 2 indagheremo sugli effetti degli ambienti di propagazione negli algoritmi di mobilità controllata; nel Capitolo 3 verranno proposte delle euristiche distribuite per filmare eventi sportivi con UAV, mentre l'approccio bio-ispirato sarà affrontato nel Capitolo 4. Infine, un nuovo protocollo di comunicazione per WSN chiamato Decentralized Time-Synchronized Channel Swapping verrà analizzato nel Capitolo 5.

Self-Organizing Systems

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 [2]. 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 [1].

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 [3]. Two more properties emerge from such definition:

- is not straightforward to guess the final pattern looking only to the local interactions (emergent behavior);

- the missing of determinism of the algorithms [4].

For these features to hold, the following are some of the conditions that must be met [3]:

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 modelled properly, could led to some kind of organization even if there is no guarantee about that [5].

Self-organization can be realized through different approaches [4]:

- *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;
 - *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;
 - *Bio-inspired methods*: biologically inspired methods build a category that is composed of neighborhood-depending operations very similar to behavior 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.

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 behavior of ad hoc networks [6]. This latter consideration allows us to consider the mobility as a fundamental aspect of the self-organizing networks.

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of low-cost, low-power, multi-functional, 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 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 micro-controller, possibilities of storing data using different type of memories and finally an electronic circuit to interface with the sensors and an energy source, usually a battery or an embedded type 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 node. 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. 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, does local processing, possibly communicates 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 constraints;
- 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) [7]. More specific details about these networks will be given in the followings subsections.

1.2 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 [8]:

- 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 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).

1.3 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 [9]:

- 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 [10];
- 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.

Impact of the Propagation Environment on Controlled Mobility Algorithms

In this chapter, we are interested in investigating the role of the propagation model used in the simulations of distributed algorithms that using the controlled mobility. This technique allow to each nodes of the network to autonomously move towards a position that is optimal in terms of energy consumption for the transmission of a data flow. In literature was been shown as the optimal placement for relay nodes that help the transmission of a data flow between a source and destination nodes is on “evenly spaced” positions on straight line between source and destination. This is true if the initial residual energies are the same for all nodes, obviously this assumption is not realistic, so the new placement for relay nodes is on “energy spaced” positions.

In the literature, the most used model for propagation is the *free-space* environment. Free-space is often too simplistic and ideal for the real propagation environment conditions. It is interesting to understand how controlled mobility works in more realistic propagation environments. Therefore, we evaluate the real benefits of controlled mobility also in the *two-ray ground* environment. The two-ray ground model refers to an environments with two-ray (as in Figure 2.1) and considers other parameters that the free-space model do not take into account. Moreover, we propose a *generic propagation model* able to predict the path loss value for different environments: this model generalizes the free space model. It is based on the same physical phenomena assumed by the free space model but introduces a *critical distance*, as a threshold on the distance between two devices. Below this threshold, the model turns to be the simple free-space model.

In order to simulate the three propagation environments presented above, we must understand how the energy consumed by the devices for communication is related to the environment by describing the energy model related to the propagation model. After the energy model has been described mathematically, we can implement it in a simulator and evaluate the results obtained.

The rest of this chapter is organized as follows: related works in Section 2.1; Section 2.2 describe the propagation environments and derive the energy model for data transfer for different environments. In Section 2.3 we show

the controlled mobility algorithms used in this work. Simulation results are presented in Section 2.4 before concluding the work in Section 2.5.

2.1 Related Works

Wireless Sensors Network (WSN) are systems composed of a set of devices capable to detect environment entities, collect data on specific events, communicate with each other, and move towards relevant positions. Certainly, one of the most important issues in the design of these networks is the energy consumption, since the devices are often powered by batteries. In this context, in recent years, the concept of mobility has been exploited also to reduce the energy consumption, by determining the best location for a device to move in order to maximize its life-time [11], [12] and [13].

The three cited works identify three macro-categories of mobility in wireless systems: random [11], predictable [12] and controlled [13] mobility. 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. In this work, we are interested in this last category of mobility.

In fact, by driving the devices in “optimal” position, we can ensure a more efficient and reliable energy management as well as provide benefits in terms of communication. For instance, it is possible to alleviate path loss, which is a key element in signal dispersion. In the free space model, path loss is proportional to the square of the distance between the devices [75]. So, the shorter the distance between two devices, the lower the signal loss and consequently the energy used for transmitting/receiving data. On the other hand, to reach the optimal positions, devices have to make physical movements that consume energy. Hence, a trade-off between the energy consumed for the movement and that used for the data transmission/reception is very important to exploit controlled mobility potentialities.

In literature, several schemes involving controlled mobility are presented for different goals, such as: coverage management [15], [16], [17], energy consumption reduction [18], [19], [20], [21], transport layer parameters improvement [22], [23], [24]. Furthermore, in [25], authors show that the virtualization of movements reduces the energy consumption.

2.2 Propagation and Energy Models

The energy consumed by a device in order to transmit 1 bit at distance d is $E = \beta \cdot d^\alpha$ where α is the path loss exponent and β is a constant [26]. In the simulations reported in literature, α is usually set to 2 as a typical value of the free-space environment [25]. A different choice in the path loss exponent leads to a different environment. Also β is related with the path loss exponent, in fact its measurement unit is [*Joule/m^α*]. In the two following subsections, we first introduce the propagation models that will be simulated and then we derive the corresponding energy model for data transmission/reception.

2.2.1 Propagation Models

The propagation models most used in literature are: the *free-space* and the *two-ray ground model*, which assign to the path loss exponent (α) a value of 2 and 4, respectively. In this work, we vary the value of α between these two extremes. To do this, we propose a *generic-model* able to use a path loss value in the range $2 \leq \alpha < 4$.

The path loss value related to the free-space model is calculated as [27]

$$PL_{free-space} = \frac{P_T}{P_R} = \left(\frac{4\pi d}{\lambda} \right)^2$$

where λ is the wavelength and d is the distance between transmitter and receiver.

Instead, the path loss value related to the two-ray ground model is [27]

$$PL_{two-ray\ ground} = \frac{P_T}{P_R} = \frac{d^4}{h_t^2 h_r^2}$$

where h_t and h_r are the transmitter and receiver antenna heights, respectively, and d is the distance between the devices as in Figure 2.1.

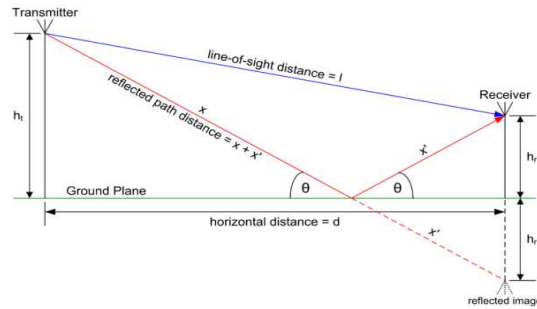


Fig. 2.1. Two-ray ground model

For all the values of α between 2 and 4, we propose a generic model able to predict the path loss value through the following relation [28], computed from the logarithmic form of the free-space path loss by using the least mean square error method [29]:

$$PL_{generic} = \left(\frac{4\pi d_0}{\lambda}\right)^2 \cdot \left(\frac{d}{d_0}\right)^\alpha + \chi$$

where d is the distance between transmitter and receiver, λ is the wavelength, χ is the value of the shadowing effect, and d_0 is the critical distance. If the distance between the devices is smaller than the critical distance, we can assume that they communicate in the free-space model. As we can see the model just described is an extension of free-space model, in fact, if $\alpha = 2$, the two models coincide. The description of these propagation models is very important because the value of the path loss exponent is an important part of the energy models that we will derive.

2.2.2 Energy Models

The following step is to derive mathematically the energy model corresponding to each mentioned propagation model.

As we already reported, the energy consumed by a device in order to send 1 bit at distance d is $E = \beta \cdot d^\alpha$, where α is the path loss exponent and β is a constant [26]. Both α and β depend on the environment: α varies between 2 and 4 according to the propagation model and β depends on the value chosen for α . Therefore, β itself changes according to the propagation environment.

The energy consumption of a generic device to send l bit at distance d , for the free-space model is:

$$E_{free-space} = l \cdot \beta \cdot d^2 = l \cdot \left\{ \frac{P_{rec}}{R} \cdot \frac{1}{G_t G_r} \cdot \left(\frac{4\pi}{\lambda}\right)^2 \right\} \cdot d^2$$

where G_t and G_r are the gains of transmitting and receiving antennas, respectively, P_{rec} is the minimum power so that a bit is received correctly, R is the transmission rate and λ is the wavelength. In order to characterize the propagation environment, we plug in this model values commonly used in Wireless Sensor Networks (WSN):

- trasmission rate $R = 10 \text{ kbits}$ (in the WSN, typically, $10 \leq R \leq 240 \text{ kbits}$);
- antenna gain $G_t = G_r = 1$;
- minimum power $P_{rec} = -85 \text{ dBm} = 3.6 \times 10^{-12} \text{ W}$;
- wavelength $\lambda = 0.125 \text{ m}$;
- operating frequency $f = 2.4 \text{ GHz}$

so we obtain the following energy model:

$$E_{free-space} = l \cdot \left\{ 3.2 \frac{pJ}{bit \cdot m^2} \right\} \cdot d^2 \quad (2.1)$$

The energy model for the two-ray ground environment casts as follows:

$$E_{two-ray\ ground} = l \cdot \beta \cdot d^4 = l \cdot \left\{ \frac{P_{rec}}{R} \cdot \frac{1}{G_t G_r} \cdot \frac{1}{h_t^2 h_r^2} \right\} \cdot d^4$$

By setting the gains of the antennas equal to 1, and antennas heights to $h_t = h_r = 1.5\ m$ [30], we obtain the following relation:

$$E_{two-ray\ ground} = l \cdot \left\{ 0.0625 \frac{fJ}{bit \cdot m^4} \right\} \cdot d^4 \quad (2.2)$$

In order to compute the energy model to use for a generic path loss exponent in the range $2 < \alpha < 4$, we set unitary gains of antennas, so to derive for free-space and two-ray ground environments the two following relations:

$$\begin{cases} E_{free-space} = l \cdot \frac{P_{rec}}{R} \cdot PL_{free-space} \\ E_{two-ray\ ground} = l \cdot \frac{P_{rec}}{R} \cdot PL_{two-ray\ ground} \end{cases}$$

therefore, according to the same logic, for generic model we will have a model equal to

$$E_{generic} = l \cdot \frac{P_{rec}}{R} \cdot PL_{generic} = l \cdot \frac{P_{rec}}{R} \cdot \left(\frac{4\pi d_0}{\lambda} \right)^2 \cdot \left(\frac{d}{d_0} \right)^\alpha$$

We decided to set the critical distance to $d_0 = 1.5\ m$ [31]

$$E_{generic} = l \cdot \left\{ 8.1862 \frac{pJ}{bit} \right\} \left(\frac{d}{1.5\ m} \right)^\alpha \quad (2.3)$$

Also in the simulation, we need to take into account the energy consumption due to the movement [32]:

$$E_m(d) = k \cdot d + E_{friction}$$

where d is the covered distance, k is a constant ($0.1 < k < 1 \frac{J}{m}$) and $E_{friction}$ is the energy needed to overcome the static friction. For our simulations k will be set to 0.1 and the static friction will be neglected since $E_{friction} \ll k \cdot d$.

2.3 Controlled Mobility Algorithms

One of the key points for this category of mobility, it is the determination of the optimal position to maximize the life-time, and thus, to minimize the energy consumption of a node during communication.

It's clear that to have less waste of energy, and therefore to maximize the life-time of a sensor, the optimum position to achieve is in the line between the source and destination nodes. Consequently, the controlled mobility drives sensors to be spaced uniformly (*evenly spaced*) in point to point communication between the transmitting node and the receiver node.

In the real world, however, it is fair to assume that the nodes are positioned with the same energy initially, but it is also proper to consider that, according to the position and the amount of data transmitted, the residual energy is different from node to node. As a result it can be shown, as done in [25], that, if the residual energy is different between the various nodes, the nodes controlled mobility will drive to a new optimal position, which will always be along the line joining the source and destination but the nodes will not be evenly spaced, but rather assume a position dependent on their residual energy (*energy spaced*). The non-uniformity in fact, derives from the fact that the residual energy is different between the nodes.

By driving the devices in “optimal” position, we can ensure a more efficient and reliable energy management as well as provide benefits in terms of communication. So, the shorter the distance between two devices, the lower the signal loss and consequently the energy used for transmitting/receiving data. On the other hand, to reach the optimal positions, devices have to make physical movements that consume energy. Hence, a trade-off between the energy consumed for the movement and that used for the data transmission is very important to exploit controlled mobility potentialities. The three energy models proposed above will be used to determine the energy consumption of different schemes of controlled mobility [24]:

- evenly-spaced (EvS);
- energy-spaced (EnS);
- virtual evenly-spaced (VEvS);
- virtual energy-spaced (VEnS).

The first scheme of controlled mobility drives nodes to space uniformly along the straight line between the source and destination nodes (Figure 2.2a), the second scheme spaces the devices along the straight line by taking into account their residual energy (Figure 2.2b). The last two controlled mobility algorithms introduce the virtual movement concept [24], *i.e.* nodes do not move until the algorithm converges to a stable point (Figure 2.2c).

All the algorithms have been tested in the three mentioned propagation models and their performance will be compared with an algorithm that does not let devices to move (*no-movement algorithm*, NM).

2.4 Results

In this section, we will show the results obtained by simulating the mentioned controlled mobility schemes through MATLAB [33] in different propagation

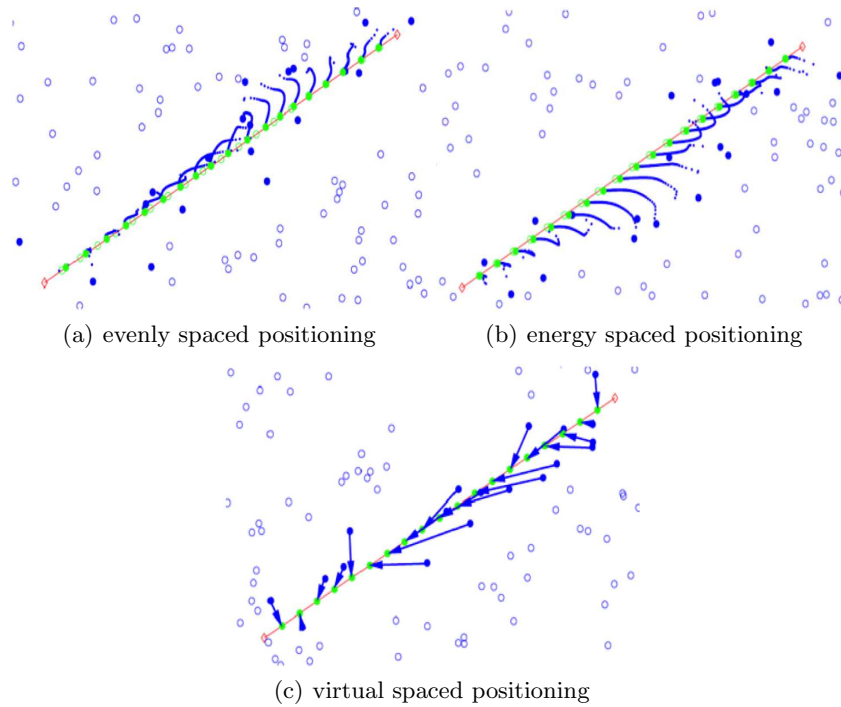


Fig. 2.2. Controlled mobility algorithms

environments. We are interested to evaluate the results in terms of residual energy of the devices after movement and data transmission. Similarly to the simulation scenario in [25], we consider a data flow between a source and a destination device in a $1000\text{ m} \times 1000\text{ m}$ field. In Table 2.1 we reported the values of the parameters used throughout the simulation. The routing issue is out of the scope of this work. Since the optimal solution for placement in terms of energy consumption requires nodes to lie on the direct route between source and destination [13], we chose to use for all the simulated algorithms a routing scheme that chooses nodes placed at the shortest distance to the straight line.

As we said in the previous section, we have tested four controlled mobility schemes: *evenly-spaced* (EvS), *energy-spaced* (EnS), *virtual evenly-spaced* (VEvS) e *virtual energy-spaced* (VEnS) [25] along with a scheme that does not allow devices to move (NM). The first scheme of controlled mobility drives nodes to space uniformly along the straight line between the source and destination nodes, the second scheme spaces the devices along the straight line by taking into account their residual energy. The last two controlled mobility algorithms introduce the virtual movement concept [25], *i.e.* nodes do not move

until the algorithm converges to a stable point. All the algorithms have been tested in the three mentioned propagation models.

2.4.1 Results of the free-space model

According to the energy model in Equation 2.1, we simulated the behaviour of five algorithms for a path loss exponent equal to 2. Specifically, we present two simulation campaigns: in the first the nodes density is fixed to $6 \cdot 10^{-4} \frac{nodes}{m^2}$ and the transmission data length varies between 0.1 and 4 Gb; in the second the size of the data to transmit is fixed to 1 Gb and the nodes density varies according to the values reported in Table 2.1.

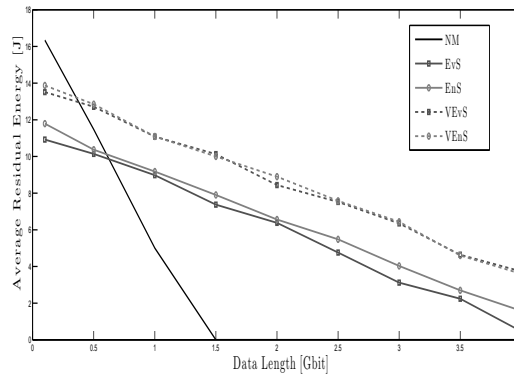


Fig. 2.3. Free-space: Residual energy vs. data sent ($\rho = 6 \cdot 10^{-4} \frac{nodes}{m^2}$)

The results of the first simulation campaign are presented in Figure 2.3 We can see that for low quantities of data sent, the algorithm without mobility presents a larger residual energy respect to the controlled mobility algorithms, but when the size of the transmission increases the gap vanishes quickly. In

Table 2.1. Simulation Parameters

Field Area ($L \times L$)	1000m \times 1000m
Source Position (x_S, y_S)	(0,0)
Destination Position (x_D, y_D)	(1000,1000)
Nodes Density (ρ)	$\{2, 4, 6, 8, 10\} \cdot 10^{-4} \frac{nodes}{m^2}$
Data Length (l)	50Mb \div 40Gb
Initial Residual Energy (E_i)	{15 \div 20}J
Max. Transmission Radius (r)	$\frac{1}{2 \cdot \sqrt{\rho}} m$
Transmission Rate (r_T)	10kbit/s
Number of run for each scenario	100

Figure 2.3 we can see that the threshold value for which is convenient to let devices move is $0.5 Gb$ for the case $\rho = 6 \cdot 10^{-4} \frac{\text{nodes}}{m^2}$.

The second simulation campaign took into consideration five scenarios with varying devices density $\rho = [2, 4, 6, 8, 10] \cdot 10^{-4} \frac{\text{nodes}}{m^2}$. Figure 2.4 shows that a higher density improves the performance in terms of residual energy after the transmission. This result was expected because a higher density of devices requires a smaller amount of movements to drive the devices to the straight line.

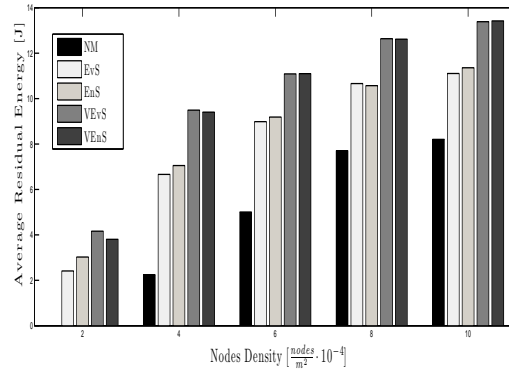


Fig. 2.4. *Free-space:* Residual energy vs. node density ($l = 1 Gbit$)

If we compare the controlled mobility algorithms, we can see that in both the simulation campaigns, the virtual movement algorithms use less energy than their corresponding real movement algorithms. This improvement is due to the energy saved by virtualizing the movement of nodes.

2.4.2 Results of the two-ray ground model

For the simulation in the two-ray ground environment, we have used the energy model in Equation 2.2. Even for this model we performed two simulation campaigns: in the first the nodes density is fixed to $6 \cdot 10^{-4} \frac{\text{nodes}}{m^2}$ and the transmission data length varies between 0.1 and $40 Gb$; in the second the size of the data to transmit is fixed to $10 Gb$ and the nodes density varies according to the values reported in Table 2.1.

From the first simulation campaign, in Figure 2.5, we can see that, even for the two-ray ground model, there is a threshold value ($7 Gb$) for the data length from which is convenient to move devices and let them find positions that are less energy consuming. One important difference in respect of the free space model is that the residual energy decreases more slowly when the transmitted data increases. This result shows the effectiveness of moving to a better placement for the devices when the propagation model is not the simple free space.

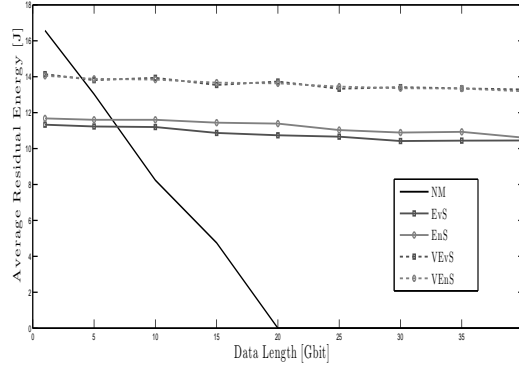


Fig. 2.5. Two-ray ground: Residual energy vs data sent ($\rho = 6 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$)

Figure 2.6 shows the average residual energy when nodes density varies. In this case, we can see that when the nodes density increases, the algorithm with no movement improves its performance and reaches almost the real movement algorithms.

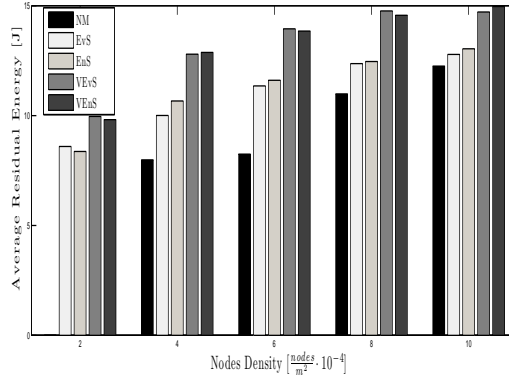


Fig. 2.6. Two-ray ground: Residual energy vs node density ($l = 10 \text{ Gbit}$)

Again the algorithms with virtualized movements perform better than their corresponding real movement ones.

2.4.3 Results of the generic model

The generic-model works with values of the path loss exponent in the range $2 \leq \alpha < 4$. Specifically, in our simulations, we have used an α variable in the range $\{2 \div 3.75\}$ with a step of 0.25, nodes density set to $\rho = 6 \cdot 10^{-4} \frac{\text{nodes}}{\text{m}^2}$ and we let the size of data vary between 1 and 300 Mb. For matter of space, in this chapter we will show only the results for $\alpha = 2.5$ and $\alpha = 3.25$.

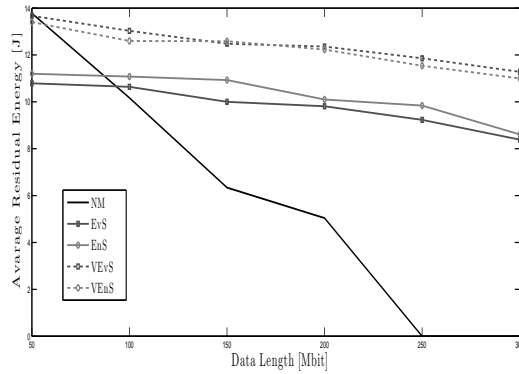


Fig. 2.7. *Generic-model:* Residual energy vs data sent ($\alpha = 2.50$)

In Figure 2.7 and 2.8, the simulation results show the same trend obtained for the previous environments: the algorithm with no movement is better than all the others, in terms of devices residual energy, until a certain data length. When $\alpha = 2.5$ the threshold on the data length is around 100 Mb in respect of the algorithms with real movements, and 50 Mb in respect of the algorithms with virtualized movements. When $\alpha = 3.25$ the threshold on the data length is around 7 Mb in respect of the algorithms with real movements, and 4 Mb in respect of the algorithms with virtualized movements.

These results reflect the behaviour observed for the previous two models and show that, because of both a faster power decay when α increases and controlled mobility robustness is very convenient to move devices and let them find the optimal energy consumption positions.

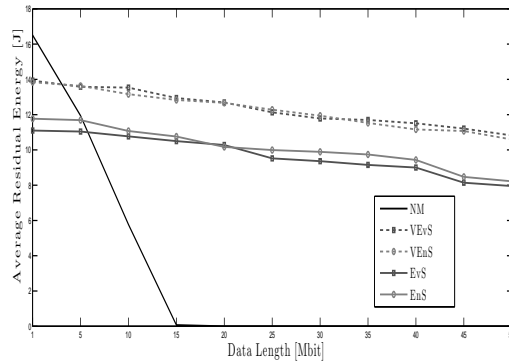


Fig. 2.8. *Generic-model:* Residual energy vs data sent ($\alpha = 3.25$)

2.5 Conclusions

In this chapter we evaluated the performance of a WSN system under different propagation models when devices use controlled mobility schemes. Five different algorithms have been tested. Four of them make devices move according to a different controlled mobility scheme, while the fifth algorithm keeps devices in the original positions.

The simulations have been performed in three propagation environments: *free-space*, *two-ray ground* and *generic* model. The first one uses a path loss exponent equal to 2, the second one that uses several more parameters to model the two-ray and a path loss exponent set to 4 and the last one able to use values of the path loss exponent in the range $2 \leq \alpha < 4$ by generalizing the free space model.

The results showed that controlled mobility leads to a lower energy consumption for all the propagation environments, when the size of the data to transmit is higher than a threshold. This threshold becomes smaller when the value of the path loss exponent increases. Furthermore, algorithms that use virtualization of the movements perform better than the corresponding real movement algorithms because of the energy saved by moving only when the algorithm converges.

This work led to the writing and publication of the following work:

R. Surace, V. Loscrí, E. Natalizio, “**On the Impact of the Propagation Environment on Controlled Mobility Algorithms**”, in *International Workshop on Mobility and Communication for Cooperation and Coordination (MC3) at IEEE ICNC 2012*.

Algorithms to Film Sport Events with Flying Robots

Flying robots, also known as Unmanned Aerial Vehicles (UAV) or drones, are aerial vehicles that operate without a human pilot. Flying robots are usually equipped with a positioning system, storage memory, and a wireless transceiver. They can fly at considerable speed, 60 km/h for commercial devices and 220 km/h for military aircrafts. Since their creation, flying robots have found many uses in civil and military applications. Currently, they are most often used for aerial reconnaissance, scientific research, logistics and transportation, or more in general, in all the situations where a direct human intervention would be hazardous. A brilliant example of flying robots' usefulness was presented in Fukushima in 2011, when a flying robot was used to explore the disaster site at Japan's devastated nuclear power plant. We are convinced that the real potential of flying robots consists in achieving coordination and cooperation among the devices of a fleet, and that the correct design of coordination/cooperation schemes would pave the way for the realization of mission-oriented devices.

In this work, we make a step in the direction of deploying coordination/cooperation schemes for flying robots operations by proposing, formulating and simulating the *Sport Event Filming* (SEF) problem. We introduce this problem in order to provide a novel application scenario, where we can develop strategies to coordinate the movement of a group of mobile robots in the presence of highly varying time-space constraints to film/monitor a sequence of actions while optimizing some specific objective. Nevertheless, a solution to this problem is of interest for several application domains. Besides TV filming, it would be beneficial for environmental monitoring, disaster recovery, site inspection and exploration, etc.

Specifically, the SEF problem copes with the organization of a fleet of flying robots able to fly over a limited field to film a sport event with the objective of maximizing the satisfaction experienced by viewers who watch the game on TV, while minimizing the traveled path.

The family of problems we deal with are usually referred to as Dynamic Vehicle Routing (DVR) problem, and the static variants taken into consider-

ation in this work are all NP-hard problems. Specifically for the event filming problem some solutions have been proposed [34], [35], [36], [37]. The main disadvantage of these solutions is that cameras are fixed. Therefore, they are not applicable to the more general problem of coordinating flying robots movements to film an event in a hostile or hazardous environment. Furthermore, they cannot provide the same level of accuracy or entertainment given by mobile devices. Whereas several solutions exist for mobile sensor networks in static scenarios, to the best of our knowledge, no schemes using flying robots have been proposed to solve this specific and dynamic problem.

The core contributions of our work can be outlined as follows:

- we describe and formulate an interesting and unexplored problem in the framework of self-organization of mobile video sensing devices;
- we propose two families of algorithms to solve the *dynamic* version of the problem in a distributed way and without any knowledge of the sequence of actions;
- we propose a mathematical model in Section 3.5 for the *static* version of the problem.

The rest of this chapter is organized as follows. Section 3.1 presents the Vehicle Routing Problem and its variants. In Section 3.3 we propose two families of distributed techniques for the optimal placement of flying robots. These schemes are tested and analyzed through several simulation campaigns in Section 3.4. Finally, in Section 3.5 we show the mathematic model used to validate the algorithms tested, before concluding the chapter in Section 3.6.

3.1 Related Works

The problem of determining the movement pattern for a certain number of flying robots when they have to film an event while maximizing viewer satisfaction and minimizing the total transportation costs can be considered as a special case of the *Vehicle Routing Problem with Time Windows* (VRPTW). Specifically, it can be classified as a *Vehicle Routing Problem with Soft Time Windows* (VRP-STW), where the sequence of points in the field to be filmed represent the customer to be served. If a specific point is not timely filmed, this affects only the satisfaction of the viewers without invalidating the overall solution.

The VRPTW assume that the cities to be visited (actions in the SEF problem) are known *a priori* and will not change during the execution of the solution. However, in real applications this assumption may be too strict. In reality, we have locations to be served that can be highly variable [39]: they can be born and die at any moment, their demands can change over time even when the solution has already been calculated. Also in the SEF problem, the position to film and the time to film that location change action by action.

The problem of planning routes through service demands that arrive during a mission execution is known as the *Dynamic Vehicle Routing Problem* (DVRP) [40], because part or all the locations to reach are not known *a priori*.

In [40], the authors identify three main approaches to address DVR problems. The first approach is to simply re-optimize every time a new event takes place; in the second approach, routing policies are designed to minimize the worst-case ratio between their performance and the performance of an optimal offline algorithm that has *a priori* knowledge of the entire input sequence; in the third approach, the routing problem is embedded within the framework of queueing theory and routing policies are designed to minimize typical queueing-theoretical cost functions such as the expected waiting time in the system for the demands.

Both families of distributed algorithms we present in this work follow the first approach. In the second family, we additionally consider specific characteristics of the problem to forecast the next locations to be covered.

3.2 Definitions and initial assumptions

To better understand the SEF problem must first be defined as a sport event is characterized and its key components: *game actions*, *drones movement* and *viewer satisfaction*.

3.2.1 Action

An action consists of a sequence of simultaneous movements performed by the main actors of the event: the ball (or football, disk, etc.), the players and the referees. The complete characterization of an action is composed of a quadruple (x, y, z, t) for each actor, where (x, y, z) are the coordinates of the actor's position in a tri-dimensional spatial reference system and t is the time instant when the actor is in that position. We assume that the camera-drones move on a plane that is fixed, parallel to the plane where the actions take place and high enough so as not to interfere with the actions. To correctly film the game, cameras will have to match the players (x, y) – *coordinates* exactly by moving over their heads and filming them from a *perpendicular* perspective. It is worth noting that in reality a perfect matching of the (x, y) – *coordinates* is not necessary for the SEF problem, because the *camera aperture diameter* can compensate for moderate errors in camera positioning. Nevertheless, we have to consider that the SEF problem is used here as a representative of a general class of coordination/cooperation problems, which could present more severe constraints for the drones, such as the need for a *perpendicular* filming perspective due to physical obstructions. Furthermore, this assumption remarkably simplifies the mathematical modeling and, to obtain a fair

comparison, it will be used also for evaluating the proposed distributed algorithms. From the camera perspective the images will be flattened on the game field plane. Hence, the z – *coordinate* for both the players and the drones can be neglected in the definition of the problem and its solution. Our future work will also consider movement of the camera over the z – *axis* or, equivalently, its zooming capabilities. A *transversal* (rather than *perpendicular*) perspective of filming, which is possible by considering the rotational capabilities of the cameras, would lead to a totally different problem and mathematical formulation that we do not consider in this work.

An action is defined within a time window, which consists of the following time instants and intervals:

- t_{birth} : is the moment when the possession of the ball is gained by another player;
- t_{start} : is the moment when the player who has possession of the ball starts moving with it;
- t_{stop} : is the moment when the player in possession of the ball loses it;
- T_{fly} : is the time interval between the loss of the ball by one player (t_{stop}) and the gain of it by another (t_{birth}).

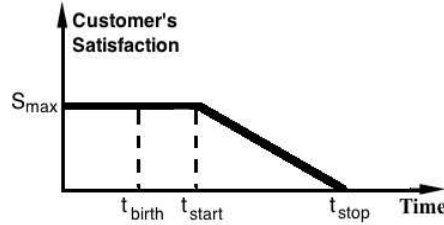


Fig. 3.1. Game action’s soft time window.

We are not interested in T_{fly} because we assume that a typical camera can follow the movement of the ball from one player to another for the whole event. Therefore, we will not use T_{fly} in optimizing the viewer satisfaction. Instead, we want to offer the viewers the possibility of enjoying the performance of the individual players involved in the event by “personalized” shots when they get possession of the ball. The duration of an action is the time interval between t_{birth} and t_{stop} . As discussed in Section 3.2.4, drones can recognize when a player is in possession of the ball. Hence, the time interval between t_{start} and t_{stop} involves all the movements of the player in possession of the ball as well as the consequent movements of the drone that is filming the current action, even when the player moves away from the ball reception position.

3.2.2 Event

An event is a sequence of actions that takes place in a size-limited field and in a predefined time span. In the following, we will use a subscript for t_{birth} , t_{start} and t_{stop} that indicates the action within the whole event.

3.2.3 Viewer satisfaction

The satisfaction S_i^k of the viewer by watching the i^{th} action of the event, filmed by the k^{th} drone is modeled as

$$\begin{cases} S_{max} & t_{arr,i}^k < t_{start,i} \\ -S_{max} \cdot \frac{t_{arr,i}^k - t_{start,i}}{t_{stop,i} - t_{start,i}} + S_{max} & t_{start,i} \leq t_{arr,i}^k \leq t_{stop,i} \\ 0 & t_{arr,i}^k > t_{stop,i} \end{cases} \quad (3.1)$$

where $t_{arr,i}^k$ is the arrival time of drone k at the position of action i . It is worth observing that the no-linear piecewise function (3.1) can be linearized by using binary variables. The related linearized function has been used when testing the proposed model. The total satisfaction of the viewer is obtained by averaging the sum of the satisfaction experienced from each action on the whole event. The satisfaction of the viewer is represented graphically in Fig. 3.1. It is maximized if the drone arrives at the action location before t_{start} and it decreases linearly until becoming 0 when $t = t_{stop}$. Hence, the definition of viewer satisfaction matches with the definition of Soft Time Window, because the only effect of a drone not arriving before t_{start} at the action location is to reduce the overall satisfaction of the viewer without invalidating the solution of the problem.

3.2.4 Relevant assumptions related to the drones

With regards to the drones movement, we refer to UAV camera-drones as any aircraft that has a limited energy E provided by its battery and capable of moving autonomously at constant speed v . We assume that our UAV camera-drones are equipped with a positioning system (GPS or indoor positioning system), a camera, storage memory and a wireless transceiver to send the filmed images to a base station and to allow communication and cooperation with other UAV camera-drones. These drones are capable of identifying and localizing a target by some radio frequency identification tag applied to it or by using a sensor network [42, 43, 44] deployed on the side of the field and capable of locating the target and communicating with the drones. Therefore, drones are able to recognize when a player is in possession of the ball because player and ball tags will be overlapping. When this occurs, the drone that is filming the action “sticks” to the player and follows him until the ball is released, and the current action terminates. Further extension of this work will consider in detail the communication aspects of filming and transmitting the images to a base station.

3.3 Distributed algorithms for dynamic VRP-STW

If the whole event sequence is available *a priori*, then the SEF problem becomes a VRP-STW problem, which has been modeled and solved to optimality [38]. Since this assumption is not realistic for a real-time system, new optimization methods need to be designed to tackle the dynamic version of the SEF problem.

In Section 3.1 we mentioned that three different approaches have been identified. The first of these approaches simply proposes to re-optimize every time a new event takes place. This approach is the most suited for the specific communication and movement capabilities of the flying robots to offer a feasible and practical solution to the event filming problem. In fact, the sub-optimal solution will be computed action-by-action by the flying robots that cooperate by exploiting their communication capabilities in a distributed and self-organized fashion. For this purpose, we introduce in the distributed strategies the *coordination time*, T_{coord} , which is the time needed by the robots to communicate with each other and determine which of them will move to follow the newly generated action.

In the following we present two families: Nearest Neighbor (NN) and Ball Movement Interception (BMI), each of them consisting of four different distributed techniques to solve the event filming problem.

3.3.1 Nearest Neighbor

The *Nearest Neighbor* technique for DVR problems in robotic system is presented in [41]. The core idea is that viewer satisfaction increases when a flying robot is able to reach the location of the current action as quickly as possible, and that the minimum traveled distance is achieved by the closest flying robot. Thus, the flying robot that is the closest to the location of the action is the one chosen to move and film the action. The following three techniques are extensions of the basic NN technique.

3.3.2 Nearest Neighbor-Division Field

A disadvantage of the NN technique is that when a sequence of actions occurs in a limited area, the same flying robots will be chosen to film it. If the duration of this sequence extends over time, it would cause one robot to reach its maximum feasible traveled distance much earlier than the others.

Based on these considerations, we introduce the *Nearest Neighbor-Division Field* (NN-DF). In the NN-DF technique, each robot is assigned to a portion of the field, and it will film the actions that are located inside that portion.

This technique has the disadvantage of not choosing the robot that is the nearest to the current action, which can result in a reduced satisfaction for the viewer. We will see in Section 3.5.3 the effects of this with respect to the reduced area for each robot to monitor.

3.3.3 Nearest Neighbor with Specular Repositioning

In the previous two techniques only one robot moves when a new action is born. The *Nearest Neighbor with Specular Repositioning* (NN-SR) technique considers robots as belonging to a pair. When one of them, k , is chosen to move to film an action for which it is the nearest neighbor, the robot that is closest to the position specular to the action position, \bar{k} , moves as well to mirror the movement of the first. More precisely, let L and W be the length and the width of the field. When robot k moves to the position of the new action (x_a, y_a) , \bar{k} will move to $(L - x_a, W - y_a)$. It is worth noting that robots are not coupled at the beginning of the event, instead \bar{k} is chosen action-by-action depending on the proximity to the action specular position. We expect that this technique, which results in robots traveling more than the previous techniques, will be more reactive and timely in filming the actions so as to offer a higher satisfaction to the viewer.

3.3.4 Nearest Neighbor with Quasi-Specular Repositioning

A generalization of the NN-SR technique is the *Nearest Neighbor with Quasi-Specular Repositioning* (NN-QSR). The NN-SR technique makes pairs of robots move specularly. As we have already highlighted, this behavior can lead to a quick depletion of the maximum allowed traveled distance, due to the specular movements of the robot (\bar{k}) that is not filming any action. Thus, the idea behind the NN-QSR is to make the center of the field be an attractor for \bar{k} while it is repositioning in the direction of k 's specular position.

The attraction strength on the movement can be modulated through an appropriate *detour factor*, $0 \leq \beta \leq 1$. When $\beta = 0$, no detour is applied on the movement of \bar{k} , which moves to the specular position in respect of the current action position, and NN-QSR coincides with NN-SR. When $\beta = 1$, \bar{k} is completely detoured towards the center of the field. For intermediate values between 0 and 1, \bar{k} move on a point on the straight line between these two extreme points. More precisely, if (x_a, y_a) , L , W are the positions of the new action, the length and the width of the field, respectively, then \bar{k} will move to $(L \cdot (1 - \frac{\beta}{2}) - x_a \cdot (1 - \beta), W \cdot (1 - \frac{\beta}{2}) - y_a \cdot (1 - \beta))$.

By detouring the movement of \bar{k} , we expect a higher satisfaction of the viewer as compared to the NN and NN-DF techniques, without introducing a high traveled distance expenditure as in the NN-SR technique.

3.3.5 Ball Movement Interception

All the previous techniques work well if t_{birth} and t_{start} are sufficiently far in time to allow a robot to reach the action location before t_{start} . In fact, these techniques try to solve the dynamic version of the proposed problem simply by adapting as quick as possible the position of one (or one pair of) robots. None of them try to forecast the location to film for next action before its

t_{birth} . As we described in [38], the static model introduces the time of “flight” of the ball when the ball is not possessed by any player, T_{fly} . This interval of time between t_{stop_i} and $t_{birth_{i+1}}$ could be used to forecast the location of next action.

We can realistically assume that robots, which are able to constantly detect the ball and its location, are also able to easily compute their trajectory. For the sake of simplicity, in this work, we consider only that the ball moves along straight lines. We consider the parabolic trajectory of the ball as flatted on the straight line lying on the game field plane, and we do not take into consideration special effects that can be given to the ball.

By assuming that robots know the trajectory of the ball, they can estimate the next player who will hold the ball. Through this estimation, before the ball reaches the next player they can start moving towards the straight line between the position of the previous action and that of the player expected to receive the ball. Thus, we introduce a new family of techniques, called *Ball Movement Interception* (BMI), which includes all the previous techniques augmented by this knowledge: *Ball Movement Interception* (BMI), *Ball Movement Interception with Division Field* (BMI-DF), *Ball Movement Interception with Specular Repositioning* (BMI-SR) and *Ball Movement Interception with Quasi-Specular Repositioning* (BMI-QSR). It is important to note that we do not assume that unexpected interceptions of the ball destined to a specific player are neglected. In fact, such events would simply cause a degradation in the performance of this family of techniques.

3.4 Performance Results

In this Section we will show two simulation campaigns illustrating selected results obtained for the proposed algorithms when several parameters vary. We consider the *average viewer satisfaction* as the output parameter for assessing the quality of the route chosen for the robots, and the *total traveled distance* as the output parameter representing the cost of the route. In the first simulation campaign we study the impact of the detour factor, β , on the performance of the Specular Repositioning techniques. The second simulation campaign is a more general comparison among the different distributed techniques. The results have been achieved by using MATLAB [33], and they have been averaged over 1000 runs with a confidence interval of 95%. The parameters presented in Table 3.5 are used in all the simulation campaigns, specific differences will be highlighted in each campaign subsection. In Table 3.1, we list the distributed algorithms names and the corresponding acronyms.

We simulate the behavior of the algorithms when the number of actions in the event and the duration of each action vary, respectively. The number of actions is useful to characterize the time-space variability of the actions in the event, whereas the duration of an action represents the dynamicity of the event. Both these input parameters depend on the kind of sport that has to be

Name	Acronym
Nearest Neighbor	NN
Nearest Neighbor with Division Field	$NN-DF$
Nearest Neighbor with Specular Repositioning	$NN-SR$
Nearest Neighbor with Quasi-Specular Repositioning	$NN-QSR_\beta$
Ball Movement Interception	BMI
Ball Movement Interception with Division Field	$BMI-DF$
Ball Movement Interception with Specular Repositioning	$BMI-SR$
Ball Movement Interception with Quasi-Specular Repositioning	$BMI-QSR_\beta$

Table 3.1. Simulated algorithms and their acronyms

filmed and their characterization is left as a future work. In our simulations, we also used a variable number of robots ($2 \div 6$), but, for matter of space, we will be able to show few results of scenarios with 2 and 4 robots.

Parameter	Value
Size of the game field ($L \times W$)	$110 \times 80 [m^2]$
Max Distance Feasible by Robots	65 [km]
Speed of Robots	15 [m/s]
Action Min Duration ($t_{birth} \rightarrow t_{stop}$)	0.2 [s]
Ball Min and Max Speed	$\{1 \div 40\} [m/s]$
Coordination Time (T_{coord})	0.2 [s]
Max Satisfaction (S_{max})	1
Actions Spatial and Temporal Distribution	random
Number of run for each scenario	1000

Table 3.2. Fixed parameters used for all simulations

3.4.1 Performance evaluation varying detour factor

In this simulation campaign we want to investigate the impact of the detour factor on the performance of the QSR techniques. Hence, we compare the results of NN-QSR and BMI-QSR, when the detour factor, β , varies in the range $\{0 \div 1\}$. We let the number of actions in the event and the duration of an action vary, as shown in Table 3.3. The range considered for the former parameter has been increased to match the time duration of a real event. The

performance of the two techniques in terms of average viewer satisfaction for different number of actions is reported in Fig. 3.2, and traveled distance for different maximum durations of the actions in Fig. 3.3.

Number of Robots	2
Robot k Position	$\{(-1)^k \frac{L}{4} + \frac{L}{2}, \frac{W}{2}\}$
Action Max Duration ($t_{birth} \rightarrow t_{stop}$)	$\{2, 6, 10\}$ [s]
Detour Factor (β)	$\{0 \div 1\}$
Number of Actions	$\{100, 500, 1000\}$

Table 3.3. Simulation parameters used for simulating NN-QSR and BMI-QSR

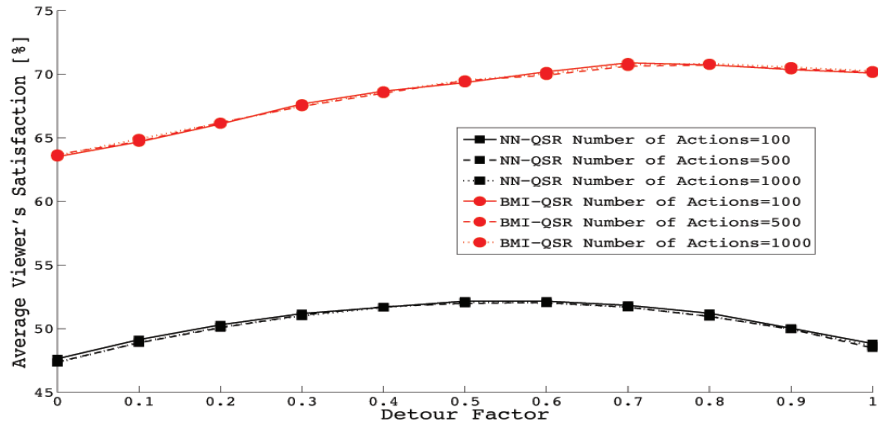


Fig. 3.2. QSR techniques: average viewer's satisfaction when the detour factor and the number of actions vary

From Fig. 3.2 we observe that use of the ball movement interception does limit the need to reposition the robot specular to the robot that is filming the action. The BMI technique leads to a high viewer's satisfaction when the attraction strength towards the center of the field increases, whereas the NN-QSR technique presents a maximum when the detour factor is between 0.5 and 0.6. This also means that different detour strengths should be applied depending on the used technique. As expected, the viewer satisfaction experienced with the BMI technique is higher on average.

The same behavior for the NN-QSR technique is presented in Fig. 3.3, where we can appreciate the existence of a minimum in the distance traveled by the robots when the detour factor is around 0.6. The BMI-QSR improves its performance when the detour factor grows until values very close to 1. It is

interesting to remark that for both the techniques, a decrease in the distance traveled by the robots corresponds to an increase in the satisfaction experienced by the viewer. Comparatively, the NN-QSR technique leads the robots to travel about 1.3 km on average less than the BMI-QSR technique, with a corresponding about 18% on average of decrease in the viewer's satisfaction. For both the output parameters, the number of actions and the action maximum duration do not significantly impact the performance of the different techniques, therefore the three simulated algorithms produce overlapping curves.

3.4.2 Comparison of Positioning Techniques

This second simulation campaign, whose main parameters are in Table 3.4, shows the results when all the distributed techniques are applied to a scenario with a variable number of actions and a fixed action maximum duration (Fig. 3.4, 3.5) and a fixed number of actions and a variable action maximum duration (Fig. 3.6, 3.7).

In Fig. 3.4 we show that the distance traveled by the robots grows linearly with respect to the number of actions for all the algorithms. Thus, it is easy to predict the distance that each technique will make robots travel through an estimate of the number of actions a real event will consist of. As expected, the NN technique is the best in terms of traveled distance, both when the Division Field is used and when it is not. The basic technique of the BMI family performs as the third best for this metric, which is a very encouraging result because of the consideration we will make about the average viewer satisfaction. On average the BMI technique makes robots travel about 73 km

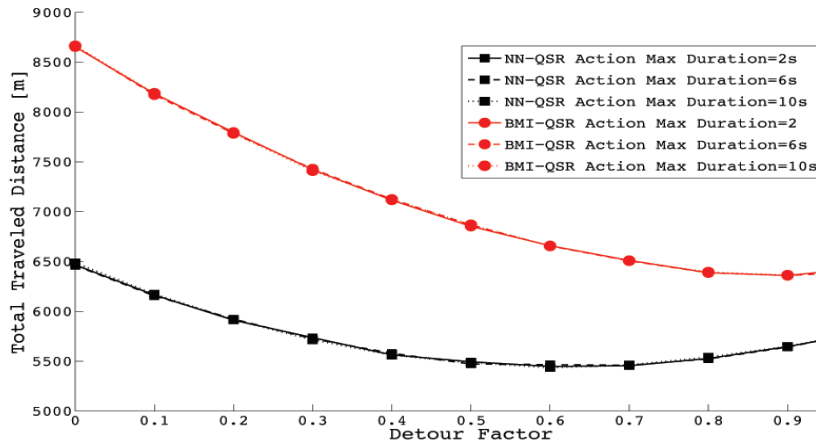


Fig. 3.3. QSR techniques: traveled distance when the detour factor and the action maximum duration vary

more than NN in the considered interval. The techniques with Specular Repositioning are the worst for this metric because of the distance traveled by the robot that does not film the action.

Number of Drones	4
Drone k Position	$\{(-1)^k \frac{L}{4} + \frac{L}{2}, (-1)^{\lceil k/2 \rceil} \frac{W}{4} + \frac{W}{2}\}$
Action Max Duration ($t_{birth} \rightarrow t_{stop}$)	$2 \div 10$ [s]
NN-QSR Detour Factor (β)	0.6
BMI-QSR Detour Factor (β)	0.8
Number of Actions	{1000 \div 5000}

Table 3.4. Simulation parameters used for distributed algorithms comparison

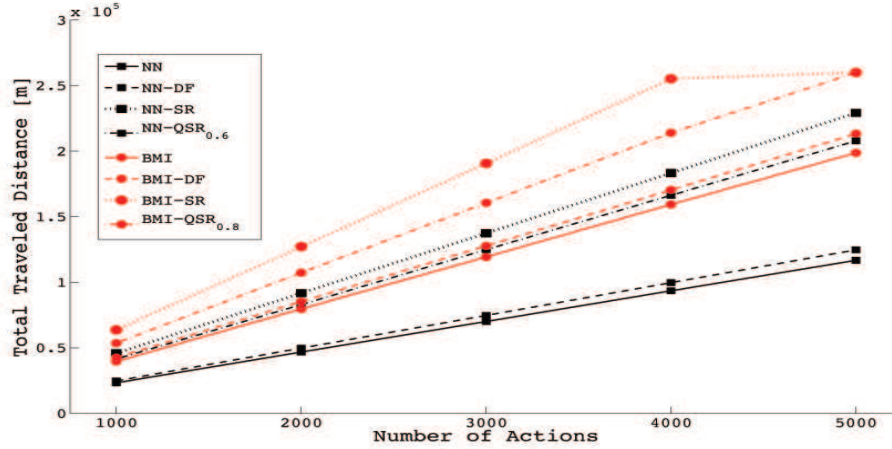


Fig. 3.4. Distributed algorithms comparison: total traveled distance for fixed actions maximum duration (6 [s])

The situation is reversed in Fig. 3.5, which shows the average viewer satisfaction. All techniques in the BMI family achieve a higher satisfaction than the corresponding techniques in the NN family. The distance between the best techniques of the two families for this parameter, which are the basic technique and the SR technique, is 14% on average. When the upper limit on the feasible traveled distance is reached, both the satisfaction achieved by BMI-SR and BMI-QSR start decreasing, since robots are not allowed to move anymore. Thus, the instantaneous satisfaction goes to zero and the average satisfaction decreases. Until the feasible distance limit is reached, the two techniques of the BMI family perform very similarly, the only main difference

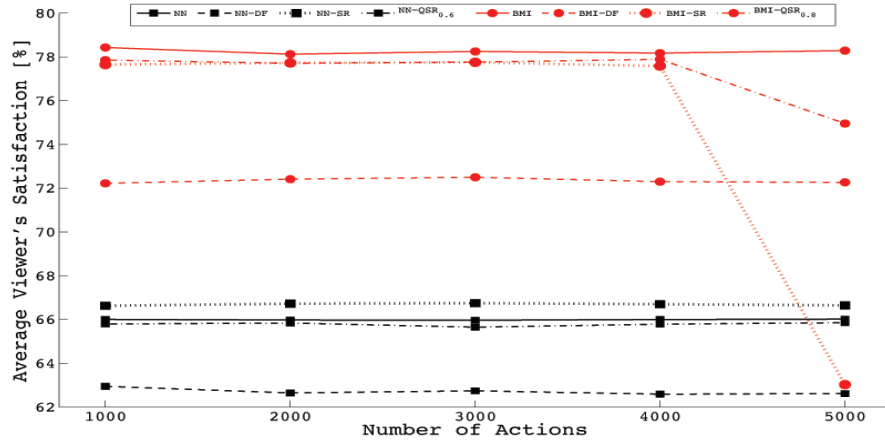


Fig. 3.5. Distributed algorithms comparison: average viewer’s satisfaction for fixed actions maximum duration (6 [s])

is that the QSR let robots travel more efficiently. Instead, we can appreciate some difference in the performance of the same techniques for the NN family, the SR technique performs 2% better on average than the QSR technique.

In Fig. 3.6 we can appreciate the traveled distance when the maximum duration of the actions varies. We can see that all the proposed algorithms are scalable with respect to this input parameter, and the heuristics ranking is the same of that in Fig. 3.4. Fig. 3.7 shows a logarithmic growth of the average viewer’s satisfaction when the actions maximum duration increases. Very quick and short actions create troubles to all the algorithms, which do not achieve more than 30% of viewer’s satisfaction, whereas they perform much better and reach 90% of satisfaction when the maximum duration is the upper value. On average, the BMI techniques have a gain of 15% over the corresponding NN techniques.

3.5 A Multi-objective Approach for UAV Problem with Soft Time Windows Constraints

In this chapter, also we give a mathematical formulation of the problem as a multi-criteria optimization model, in which the total distances traveled by the UAVs (to be minimized), the customer satisfaction (to be maximized) and the number of used UAVs (to be minimized) are considered simultaneously. A dynamic variant of the basic optimization model, defined by considering the rolling horizon concept, is shown. In order to test this model, we introduce a case study as an application scenario, where sport actions of a football match are filmed through a distributed UAVs system. The customer satisfaction and

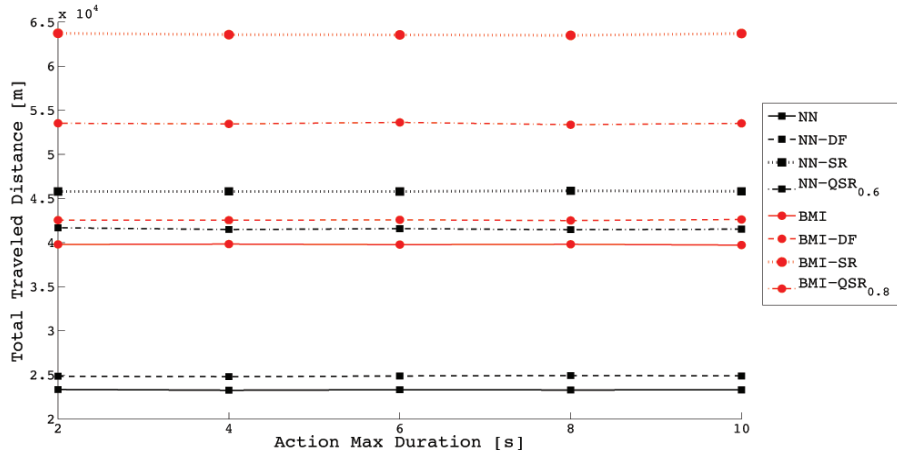


Fig. 3.6. Distributed algorithms comparison: total traveled distance for a fixed number of actions (1000)

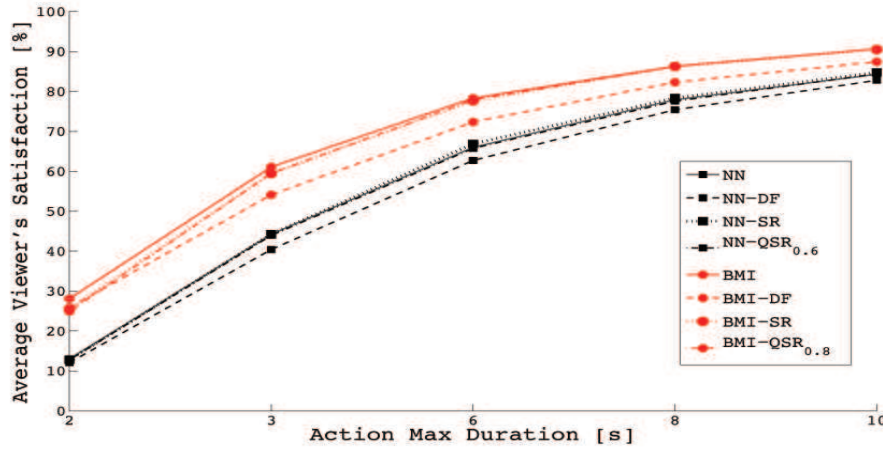


Fig. 3.7. Distributed algorithms: average viewer's satisfaction for a fixed number of actions (1000)

the traveled distance are used as performance parameters to evaluate the proposed approaches on the considered scenario.

3.5.1 Problem Statement and Mathematical Formulation

The considered scenario is characterized by a set of UAVs that fly over a finite dimension area in order to reach all the events, which occur in a finite time horizon. The satisfaction degree achieved by the customer is determined by

considering the instant of the time that a drone reaches a particular event location. Indeed, it is a measure of the temporal event coverage.

Modeling through graph theory

The UAVs single-objective routing problem has been introduced in [45]. In this section, we consider the trade-off among three conflicting objectives, namely the total traveled distance, the customer's satisfaction and the number of used UAVs. The mathematical model presented in this work can be viewed as multi-criteria extension of the model in [45] and it can be described by using the graph theory as follows.

Let $G = (V, A)$ be a directed graph where $N = \{1 \dots n\}$ is the vertices set and A is the arcs set. The vertices i with $i = (m + 1) \dots (n - 1)$, where m is the number of available UAVs, denote the events to be reached and monitored, these events contribute to the customer satisfaction evaluation, whereas the vertices j with $j = 1 \dots m$ identify the drones starting position and n represents the final event.

A non-negative cost d_{ij} is associated to each arc $(i, j) \in A$, representing the Euclidean distance from vertex i to vertex j . It is assumed that $d_{ii} = \infty$ with $i = 1 \dots n$, that is loops on the same event are prohibited.

Let us consider a set of m ($0 < m < n$) identical drones, constrained by a maximum allowed traveled distance d_{max}^k , $k = 1, \dots, m$ initially positioned at the vertices j with $j = 1 \dots m$, and able to move in two dimensions with constant and homogeneous speed v . Each drone must follow at most one route, starting from its initial position, including a set of events and ending to n . Each event should be reached by exactly one drone.

Each vertex is associated with a time window, in which the corresponding event i remains active $[t_{birth,i}; t_{stop,i}]$ where $i = (m + 1) \dots n$. A drone that reaches an event must stay in the event position until the corresponding time window ends. Each event is associated with a satisfaction function (Fig. 3.1) and the instant of time in which a drone starts monitoring an event influences the satisfaction obtained by the customer.

The goal is to find a route to be followed by each drone such that all the events are monitored and some criteria are optimized simultaneously.

Notations and Definitions

In order to describe the proposed mathematical model it is useful to introduce the following notations and definitions.

- $A = L \times W$ size-limited area;
- $[0 \dots T]$ time horizon;
- $N = \{1 \dots n\}$ set of events spatially distributed in A and temporally distributed in $[0 \dots T]$;

- $M = \{1 \dots m\}$ drones (vehicle) able to move in 2 dimensions with constant and homogeneous speed v ($0 < m < n$);
- d_{max}^k maximum feasible distance traveled by vehicle k ;
- $1, \dots, m$ drones initial positions;
- n drones final position;
- $t_{birth,i}$, $t_{start,i}$ and $t_{stop,i}$ born, start and stop time instant of the event i $\forall i \in N \setminus \{1, \dots, m\}$;
- $t_{birth,i} < t_{start,i} < t_{stop,i} \forall i \in N \setminus \{1, \dots, m\}$;
- $t_{stop,i} < T \forall i \in N \setminus \{1, \dots, m \text{ et } n\}$ and $t_{stop,n} = T$ (hypothesis in order to conclude the events before the scenario end);
- d_{ij} Euclidean distance between event i and $j \forall i, j \in N$;
- $t_{arr,i}^k$ arrival time instant of vehicle k to event $i \forall i \in N \setminus \{1, \dots, m\}$ and $\forall k \in M$;
- $t_{dep,i}^k$ departure time instant of vehicle k from event $i \forall i \in N \setminus \{n\}$ and $\forall k \in M$;
- $t_{i \rightarrow j}^k = \frac{d_{ij}}{v}$ time required by vehicle k to go from event i to $j \forall i, j \in N$ and $\forall k \in M$;
- x_{ij}^k binary variable used to indicate if drone k travels along the arc i - j

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ travels along arc (i,j)} \\ 0 & \text{otherwise} \end{cases}$$

- y_i^k binary variable to indicate if drone k reached the event i

$$y_i^k = \begin{cases} 1 & \text{if vehicle } k \text{ reaches the event } i \\ 0 & \text{otherwise} \end{cases}$$

- S_i^k customer satisfaction achieved when the event i is reached by drone k $\forall i \in N \setminus \{1, \dots, m \text{ et } n\}$ and $\forall k \in M$. The mathematical description is given in (3.1);
- S_{max} max satisfaction obtainable by the customer in a single event;
- $\Psi = \sum_{k=1}^m \sum_{i=m+1}^{n-1} S_i^k$ total customer satisfaction, *i.e.* the sum of customer satisfaction perceived in all events;
- $\bar{\Psi} = \frac{\sum_{k=1}^m \sum_{i=m+1}^{n-1} S_i^k}{n-m-1}$ average customer satisfaction;
- $\bar{\Psi}_{\%} = \bar{\Psi} \times 100$ percentage average customer satisfaction;
- Ψ_{min} minimum level of ensured satisfaction;
- $r_k = (v_1, v_2, \dots, v_h)$ with $v_1 \in \{1, \dots, m\}$, $v_h = n$, $v_2 \dots v_{h-1} \in N \setminus \{1, \dots, m \text{ et } n\}$ and $\forall k \in M$. Each drone has to travel a route that starts from its initial position and finish at final position.

It is worth observing that a piecewise linear function is used to represent the customer's satisfaction, given in (3.1). In order to linearize this function, the following variables are introduced:

- $\delta_{1i}^k, \delta_{2i}^k$ and δ_{3i}^k binary variables defined as follows:

$$\delta_{1i}^k = \begin{cases} 1 & \text{if } t_{arr,i}^k \leq t_{start,i} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{2i}^k = \begin{cases} 1 & \text{if } t_{start,i} < t_{arr,i}^k \leq t_{stop,i} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{3i}^k = \begin{cases} 1 & \text{if } t_{stop,i} < t_{arr,i}^k \leq t_{stop,n} \\ 0 & \text{otherwise} \end{cases}$$

- z_{1i}^k, z_{2i}^k and z_{3i}^k non-negative variables.

The function (3.1) for each $i \in N \setminus \{1, \dots, m\}$ and for each $k \in M$ assumes the following form:

$$S_i^k = S_{max} \delta_{1i}^k + S_{max} \delta_{2i}^k + \left(\frac{-S_{max}}{t_{stop,i} - t_{start,i}} \right) z_{2i}^k$$

with the variables constrained as follows:

$$\begin{aligned} \delta_{1i}^k + \delta_{2i}^k + \delta_{3i}^k &= 1 \\ 0 &\leq z_{1i}^k \leq t_{start,i} \cdot \delta_{1i}^k \\ 0 &\leq z_{2i}^k \leq (t_{stop,i} - t_{start,i}) \cdot \delta_{2i}^k \\ 0 &\leq z_{3i}^k \leq (t_{stop,n} - t_{stop,i}) \cdot \delta_{3i}^k \end{aligned}$$

Assumptions

- The time and spatial distribution of events are known in advance;
- All the events (except the last one) must be reached by exactly 1 drone;
- Drones start from different positions;
- Drones, that accomplished their tasks, converge towards a dummy location (last event), where maintenance operations on the vehicles can be performed. The distance traveled to reach this final position is not taken into account in the total cost evaluation;
- All events $\in N \setminus \{1, \dots, m\}$ have their own soft time window already presented in Fig. 3.1.

Mathematical model

The UAVs routing problem with soft time windows has been mathematical represented by defining a multi-criteria optimization model. The considered

objective functions are related to the following three specific aspects: minimize traveled distance, maximize average customer satisfaction and minimize the number of used vehicles.

In the evaluation of the first two objectives, the events corresponding to the initial drones' position and to the last dummy position are not taken into account. The third criterion is determined by considering the number of vehicles that reach the last event directly from their initial position, without reaching other events. These UAVs represent the vehicles that are not used.

The proposed formulation can be mathematically represented as follows:

$$\min \sum_{k=1}^m \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} d_{ij} \cdot x_{ij}^k \quad (3.2)$$

$$\max \bar{\Psi} \quad (3.3)$$

$$\min \left(m - \sum_{k=1}^m \sum_{i=1}^m x_{in}^k \right) \quad (3.4)$$

Subject to:

$$\sum_{k=1}^m y_i^k = 1 \quad \forall i \in N \setminus \{n\} \quad (3.5)$$

$$\sum_{k=1}^m y_n^k = m \quad (3.6)$$

$$y_i^k \geq x_{ji}^k \quad \forall i, j \in N, k \in M \quad (3.7)$$

$$y_i^k \geq x_{ij}^k \quad \forall i, j \in N, k \in M \quad (3.8)$$

$$\sum_{i=1}^m \sum_{j=m+1}^n x_{ij}^k = 1 \quad \forall k \in M \quad (3.9)$$

$$\sum_{i=1}^{n-1} x_{in}^k = 1 \quad \forall k \in M \quad (3.10)$$

$$\sum_{j=1}^{n-1} \sum_{k=1}^m x_{ji}^k = 1 \quad \forall i \in N \setminus \{\{1, \dots, m\} \cup \{n\}\} \quad (3.11)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ij}^k = 1 \quad \forall i \in N \setminus \{n\} \quad (3.12)$$

$$\sum_{i=1}^n x_{iz}^k - \sum_{j=1}^n x_{zj}^k = 0 \quad \forall z \in N \setminus \{\{1, \dots, m\} \cup \{n\}\}, k \in M \quad (3.13)$$

$$x_{ii}^k = 0 \quad \forall i \in N, k \in M \quad (3.14)$$

$$\sum_{i=1}^m \sum_{j=1}^m x_{ij}^k = 0 \quad \forall k \in M \quad (3.15)$$

$$t_{arr,i}^k = 0 \quad \forall i \in \{1, \dots, m\}, k \in M \quad (3.16)$$

$$t_{arr,j}^k = \sum_{i=1}^n (t_{dep,i}^k + t_{i \rightarrow j}^k) \cdot x_{ij}^k \quad \forall j \in N \setminus \{1, \dots, m\}, k \in M \quad (3.17)$$

$$t_{dep,i}^k \geq 0 \quad \forall i \in N, k \in M \quad (3.18)$$

$$t_{dep,i}^k \geq t_{stop,i} \cdot y_i^k \quad \forall i \in N \setminus \{\{1, \dots, m\} \cup \{n\}\}, k \in M \quad (3.19)$$

$$t_{dep,i}^k \leq t_{stop,n} \cdot y_i^k \quad \forall i \in N, k \in M \quad (3.20)$$

$$t_{dep,n}^k = t_{stop,n} \quad \forall k \in M \quad (3.21)$$

$$t_{arr,n}^k \leq t_{stop,n} \quad \forall k \in M \quad (3.22)$$

$$\delta_{1i}^k + \delta_{2i}^k + \delta_{3i}^k = y_i^k \quad \forall i \in N, k \in M \quad (3.23)$$

$$0 \leq z_{1i}^k \leq t_{start,i} \cdot \delta_{1i}^k \quad \forall i \in N, k \in M \quad (3.24)$$

$$0 \leq z_{2i}^k \leq (t_{stop,i} - t_{start,i}) \cdot \delta_{2i}^k \quad \forall i \in N, k \in M \quad (3.25)$$

$$0 \leq z_{3i}^k \leq (t_{stop,n} - t_{stop,i}) \cdot \delta_{3i}^k \quad \forall i \in N, k \in M \quad (3.26)$$

$$S_i^k = S_{max} \delta_{1i}^k + S_{max} \delta_{2i}^k + \left(\frac{-S_{max}}{t_{stop,i} - t_{start,i}} \right) z_{2i}^k \quad \forall i \in N \setminus \{1, \dots, m\}, k \in M \quad (3.27)$$

$$S_i^k = 0 \quad \forall i \in \{1, \dots, m\}, k \in M \quad (3.28)$$

$$t_{arr,i}^k = t_{start,i} \cdot \delta_{2i}^k + t_{stop,i} \cdot \delta_{3i}^k + z_{1i}^k + z_{2i}^k + z_{3i}^k \quad \forall i \in N, k \in M \quad (3.29)$$

The tree objective functions represent the total traveled distance (3.2) to be minimized, the average customer satisfaction (3.3) to be maximized and the number of used vehicles (3.4) to be minimized.

Each event (except the final one) must be monitored by exactly one drone (3.5), while the last event must be reached by all the drones (3.6). The event j must be monitored by the drone that followed the path that goes from event i to j (3.7-3.8).

All vehicles must start from their initial position and stop in the final event position (3.9-3.10). For each event, exactly one path entering and outgoing from it must be present in the final solution (3.11-3.13). The drones cannot follow loops, *i.e.* they cannot return to an event previously monitored (3.14) and cannot reach the other drones' initial positions (3.15). The arrival time to the initial position is set equal to the starting simulation instant (3.16), instead, the arrival time to the event j (excluding the initial event) by the vehicle k must be equal to time of departure from event i added to the time it takes to go from i to j (3.17).

A drone can leave an event only after the simulation time is started (3.18), the event work is finished (3.19) and before the end of the simulation time horizon (3.20).

A drone must never leave the last event, therefore, the departure from final event has only a symbolic value equal to the end of the simulation (3.21) and a vehicle must reach the final position before the end of the simulation period (3.22). The constraints (3.23-3.29) are used to linearize the customer's satisfaction condition (3.1).

Case Study: Sport Event Filming Problem

In section 3.5.1, we modeled the UAVs movement in order to make them able to reach a set of events and stay in these positions until the end of the event itself.

The proposed model is useful for representing many real applications [46] [47] [48]. In this work, we consider the *Sport Event Filming* (SEF) problem as a case study.

In this context, the challenge is to organize a fleet of drones able to fly over a limited field and film a sport event with the objective of maximizing the satisfaction experienced by customers who watch the game on TV.

In order to apply the proposed model to the SEF problem, we assume that a match is the scenario in which a set of game actions (*i.e.* the events of the generic model) is randomly deployed in space-time PoI.

An example of game action is given in Fig. 3.8, where, in subfigure 1, the ball reaches the player 1 and a new game action borrows. In subfigure 2, player 1 starts its action by moving and in subfigure 3 player 1 stops its action when he loses the ball, passing it to the player 2. In subfigure 4 the ball reaches player 2 and a new action borrows.

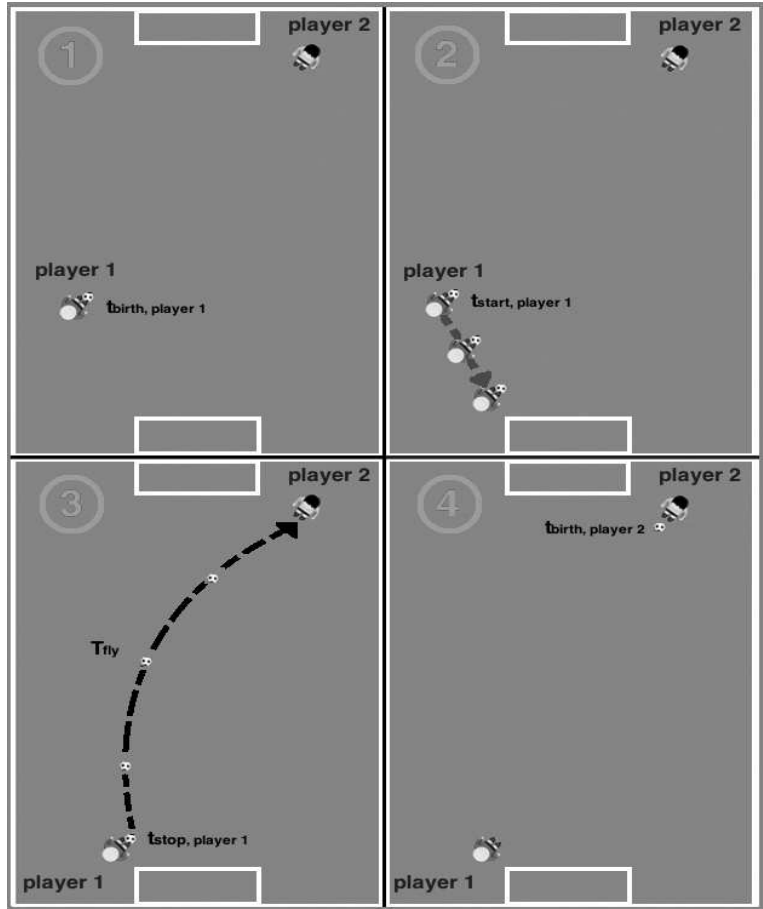


Fig. 3.8. Example of game action

We will not use T_{fly} in optimizing the spectators satisfaction because we assume that an usual camera can follow the movement of the ball from one player to another for the whole event. The goal is to compute the movements of a drones set equipped with a camera, to achieve all the game actions and film them.

An additional requirement in order to adapt the generic mathematical model to the SEF problem is: the events (game actions for SEF problem) must be sequential and not occur simultaneously, in fact the couple player-ball are in only one time-space location at each time and game actions are serial. To ensure these requirements, we set $t_{stop,i} < t_{birth,j} \forall i, j \in N \setminus \{1, \dots, m\}$ with $i < j$. In addition, the constraints (3.14) are modified as follows:

$$\sum_{j=1}^i x_{ij}^k = 0 \quad \forall i \in N, k \in M \quad (3.30)$$

With the introduction of these constraints, the actions are causal in the time, that is drones cannot reach an action before the action occurs and are forbidden to produce loops.

3.5.2 The solution approaches

In this section, the solution approaches proposed to address the multi-objective UAVs routing problem are described in details.

Two different solution strategies have been defined to address the UAVs routing problem. The former (i.e., the ϵ -constraint method) assumes that all the events are known in advance and allows to determine an approximation of the Pareto front, the latter is a rolling horizon strategy. In what follows, the proposed methods are described in details.

The ϵ -constraint method

Several approaches for solving multi-objective optimization problems have been proposed in the scientific literature [49]. In this work, in order to determine the set of efficient solutions, the ϵ -constraint method, introduced in [50], is applied.

The main idea of this method is to select only one of the objective functions to be optimized, whereas all others are converted into constraints. Thus, a set of ϵ -constraint problems $P_i(\epsilon)$, one for each objective $i = 1 \dots k$ at a time is solved.

The ϵ -constraint method is applied to the bi-objective version of the UAVs routing problem, where the total distance traveled and the customer's satisfaction are taken into account. The third objective is tackled as a parameter of the optimization procedure, in the sense that the number of UAVs to be used is fixed at each iteration of the overall algorithm. However, it is important to point out, that since m varies within the range of meaningful values that can be assigned to the number of UAVs, the overall optimization process allows us optimizing all the three objectives simultaneously.

Thus, let \bar{m} a given number of available drones, the following two optimization problems (i.e., $P_1(\epsilon_2)$ and $P_2(\epsilon_1)$) are solved iteratively.

$$\text{minimize } \sum_{k=1}^{\bar{m}} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} d_{ij} \cdot x_{ij}^k \quad (3.31)$$

subject to:

$$x \in X \quad (3.32)$$

$$\bar{\Psi} \geq \epsilon_2 \quad (3.33)$$

$$\text{maximize } \bar{\Psi} \quad (3.34)$$

subject to:

$$x \in X \quad (3.35)$$

$$\sum_{k=1}^{\bar{m}} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} d_{ij} \cdot x_{ij}^k \leq \epsilon_1 \quad (3.36)$$

where X denotes the feasible region defined by the constraints (3.5-3.13), (3.15-3.29) and (3.30).

Thus in the first model we optimize the total traveled distance and we take into account the customer's satisfaction as an ϵ -constraint; in the latter, the customer's satisfaction is optimized and the total traveled distance is handled as an ϵ -constraint. At each iteration, the value of the parameters ϵ_1 and ϵ_2 , are adequately modified.

A rolling horizon strategy

We defined a rolling horizon approach in order to capture the dynamicity of the considered scenario. In the static case, it is assumed that all of the events are known in advance, instead in the dynamic case this assumption is relaxed: events can start at any time of the considered time horizon.

In order to handle this specific situation, the route to be followed by the drones are planned by assuming the availability of partial known information about the position and the instant of time in which each event takes place [51]. In particular, let n be the number of events to be monitored, it is assumed that only a certain number r of events ($0 < r < n$) is known at each decision epoch. Thus the proposed static model is used to define the best UAVs routing, by considering only the known events and no information on future events is considered.

When a new set of r events become available (i.e., in the subsequent decision epoch), the new routing is determined by considering as initial drones' positions those obtained in the previous optimization.

3.5.3 Computational Experiments

The computational experiments have been carried out on Hewlett-Packard m9460it, Intel Core 2 Quad Q9400 2.66 GHz and 4 GB Ram with operating system Windows Vista 64 bit.

To solve the proposed mathematical model, we used LINGO 9.0 ([52]), a tool designed to build and solve different optimization models in efficient way.

In order to assess the behavior of the considered solution approaches, the SEFP as been considered as a case study. In particular, the specific scenario, whose main characteristics are reported in Table 3.5, has been considered in the computational phase.

Parameter	Value
Size of the game field ($L \times W$)	$110 \times 80 [m^2]$
Action Min/Max Duration ($t_{birth} \rightarrow t_{stop}$)	$\{0.2 \div 6\} [s]$
Number of game actions	20
Ball Min/Max Speed	$\{1 \div 40\} [m/s]$
Coordination Time (T_{coord})	$0.2 [s]$
Max Satisfaction (S_{max})	1
Actions Spatial and Temporal Distribution	random
Number of run for each scenario	1000
Confidence interval of	95%

Table 3.5. Values of the relevant parameters used for the experimental testing

The number of drones \bar{m} has been varied in the interval $[1, \dots, 6]$. We considered 6 as the maximum number of drones since the application scenario is based on a sport event whose field size is small. In fact, we will observe to the follow that difference of performance is not so appreciable when we pass from 5 to 6 drones. The set of efficient solutions obtained by the ϵ -constraint method is depicted in Fig. 3.9.

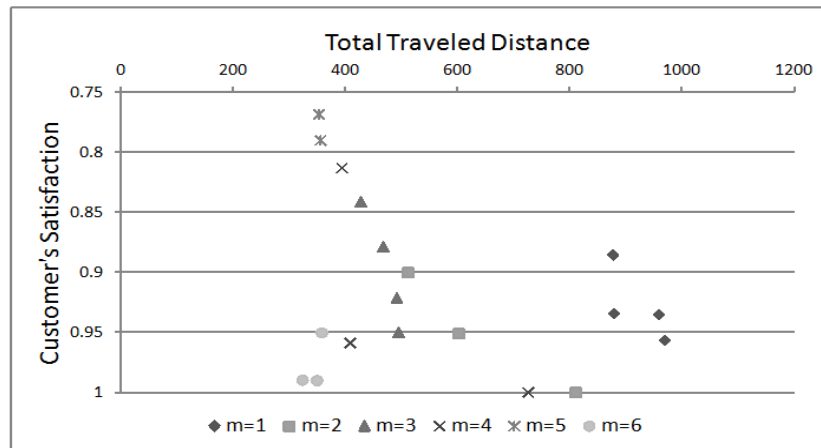


Fig. 3.9. Graphical representation of the Pareto front obtained with the ϵ -constraint method

In order to show the effectiveness of the proposed solution approach, we have also carried out computational experiments by treating the problem as a single-objective optimization problem, where a convex combination of the three objective functions has been considered.

The related problem assumes the following form:

$$\text{minimize } \alpha \sum_{k=1}^{\bar{m}} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} d_{ij} \cdot x_{ij}^k - \beta \bar{\Psi} + \gamma \left(\bar{m} - \sum_{k=1}^{\bar{m}} \sum_{i=1}^{\bar{m}} x_{in}^k \right) \quad (3.37)$$

subject to:

$$x \in X \quad (3.38)$$

where α , β and γ are non-negative parameters chosen in such a way that $\alpha + \beta + \gamma = 1$.

To generate non-dominated solutions, the single-objective optimization problem has been solved for different values of these parameters and different values of the number of drones \bar{m} . Also in this case \bar{m} has been chosen in the interval $[1, \dots, 6]$, whereas α , β and γ have been selected as in Table 3.6. The related results are reported in Fig. 3.10. By observing Fig. 3.10, the advantage of the use of the ϵ -constraint method is evident. The superiority of this approach is underlined by the results reported in what follows.

α	β	γ
1	0	0
0	1	0
0	0	1
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
1	1	1
$\frac{1}{2}$	$\frac{1}{2}$	0
$\frac{1}{2}$	0	$\frac{1}{2}$
0	$\frac{1}{2}$	$\frac{1}{2}$

Table 3.6. Values for the parameters α , β and γ

More specifically, in order to compare the two considered solution approaches, the quality of the Pareto approximation set is evaluated by considering the diversity of the set. In particular, the spacing metric proposed by Schott in [53] is used. The aim of this metric is to evaluate how evenly the points in the approximation set are distributed in the objective space. It is defined as follows:

$$\Delta = \sqrt{\frac{1}{\eta - 1} \sum_{i=1}^{\eta} (\bar{d} - d_i)^2}$$

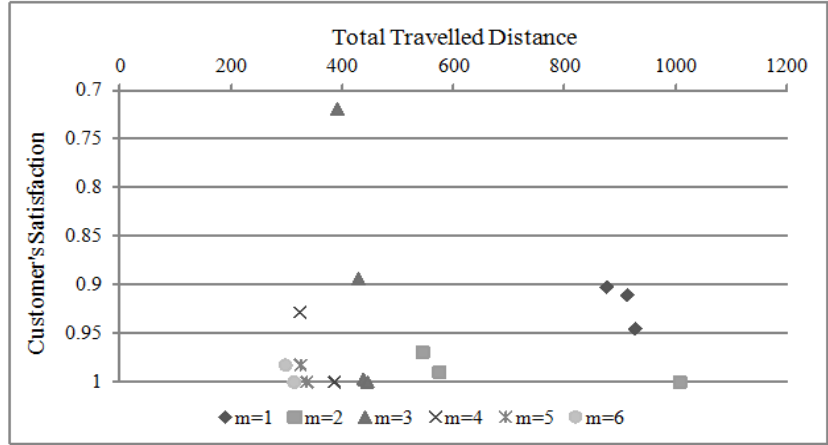


Fig. 3.10. Graphical representation of the Pareto front obtained by solving the single-objective optimization

where $d_i = \min_{j=1, \dots, \eta, j \neq i} |f_1^i - f_1^j| + |f_2^i - f_2^j| + |f_3^i - f_3^j|$, f_1, f_2, f_3 represent the considered criteria and \bar{d} is the mean of d_i , $i = 1, \dots, \eta$.

The spread metric S introduced in [54] has been also considered. This metric is used to evaluate if the set of solutions obtained span the entire Pareto optimal region and it is defined as follows:

$$S = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{\eta-1} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + (\eta - 1)\bar{d}}$$

where d_m^e represents the Euclidean distance between the extreme solutions of Pareto optimal front and the boundary solutions of the obtained non-dominated set corresponding to m -th objective function; d_i denotes the Euclidean distance between neighboring solutions in the obtained non-dominated solutions set and \bar{d} is the mean value of these distances. The smaller the value of S , the better the diversity of the nondominated set.

It is worth observing that this metric works only for bi-objective optimization problems. Thus, in order to evaluate S , we have considered only two of the three objective functions that is the total traveled distance and the customer's satisfaction.

The obtained results are given in Table 3.7, they underline that for the considered scenario the ϵ -constraint method outperform the single objective optimization approach.

Three different scenarios have been considered to evaluate the performance of the proposed rolling horizon approach, obtained by varying the value of the parameter r , representing the number of events known at each decision epoch. In particular, r has been set equal to 3, 4 and 5. In addition, at each

	ϵ -constraint Method	Single Objective Optimization
S	19.08	27.50
Δ	1.00	1.13
Number of Pareto Solutions	19	15

Table 3.7. Values of spacing and spread metrics and number of efficient points for the ϵ -constraint method and the single objective optimization approach

time instant of the rolling horizon, for a given number of drones, we solve a single-objective optimization model in which the total distance traveled by the drones is minimized and the customer’s satisfaction is handled as a constraint. The number of drones m has been set equal to 1, 2, 3, 4, 5 and 6, whereas the lower limit on the customer’s satisfaction has been chosen equal to 0.50, 0.90. The case in which $\bar{\Psi}$ is not constrained has been also considered.

The related results are given in Fig. 3.11, they clear underline that the best performance are obtained when $r = 3$.

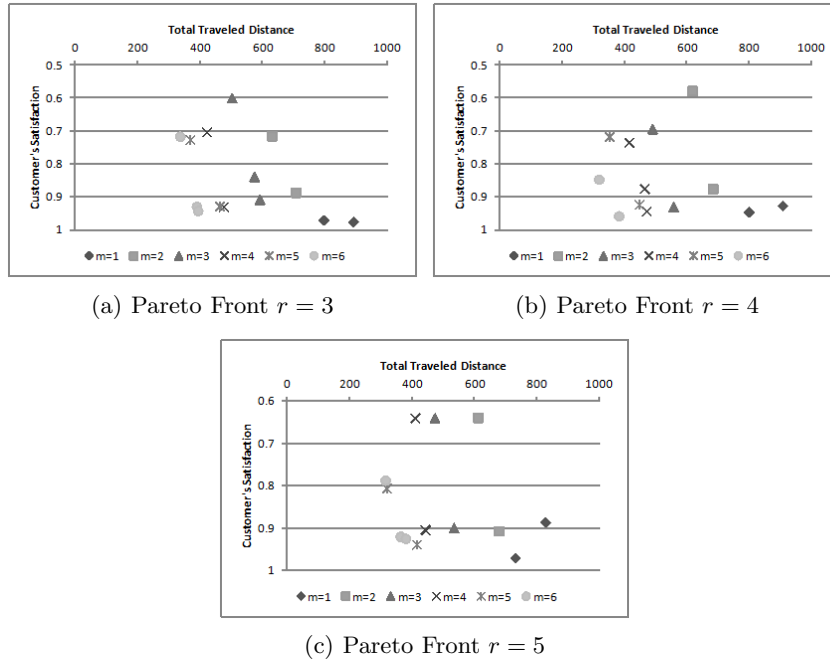


Fig. 3.11. Pareto Front determined by the rolling horizon approach

This behavior can be explained by observing that the lower r , the higher the number of the times in which the model is solved and thus the better is the representation of the dynamicity of the problem.

In order to evaluate the performance of the proposed rolling horizon strategy, we have compared this approach with the heuristic techniques proposed in [45], where the UAVs routes are re-optimized every time a new action starts. The sub-optimal solution is computed action-by-action by the drones that cooperate by exploiting their communication capabilities in a distributed and self-organized fashion.

The related results are given in Tables 3.8, where for each heuristic and for each number of the available drones, indicated with \bar{m} , the sum of the distance traveled by each UAV (i.e. the total distance) and the average customer's satisfaction are given.

From the results we can argue as, in terms of satisfaction, BMI-based techniques generally behave better than techniques without BMI, when the number of nodes is smaller. In fact, the gap in terms of satisfaction level is around 30% when only a UAV is considered and $\approx 8\%$ when the number of drones is 3. This better level of satisfaction is paid in terms of total distance traveled. In fact, the difference in terms of distance between the BMI-based and no-BMI-based approaches can achieve values greater than 500 meters when the number of UAVs is smaller than 3. When the number of devices increases (≥ 4), we can notice as performance behaviors in terms of both satisfaction and total distance between the two macro-class of approaches (i.e. BMI-based and no-BMI-based) decreases. These considerations allow us to conclude that BMI-approaches are preferable when the number of available devices is smaller.

The set of efficient solutions determined by applying all the considered heuristics is given in Fig. 3.12.

From the collected results, it is evident that, for the considered scenario, the rolling horizon approach behaves the best. Indeed, the solutions determined by this last strategy dominates those identified by the heuristics.

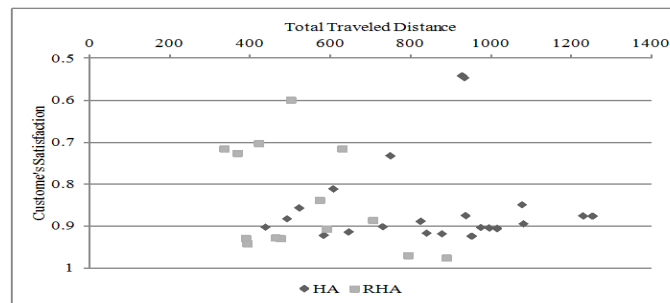


Fig. 3.12. Pareto Front determined by heuristic approaches

Heuristic	\bar{m}	Total Distance (meter)	Customer's Satisfaction	\bar{m}	Total Distance (meter)	Customer's Satisfaction
NN	1	931.91	0.55	2	748.57	0.73
NN-SR	1	926.85	0.54	2	1231.14	0.75
NN-QSR _{0.5}	1	937.23	0.54	2	1041.40	0.78
NN-QSR _{0.6}	1	929.17	0.54	2	1036.66	0.78
NN-QSR _{0.7}	1	935.73	0.54	2	1043.81	0.77
BMI	1	1076.62	0.85	2	935.72	0.87
BMI-SR	1	1084.90	0.84	2	1667.66	0.86
BMI-QSR _{0.6}	1	1087.39	0.84	2	1275.02	0.87
BMI-QSR _{0.7}	1	1079.75	0.84	2	1252.38	0.88
BMI-QSR _{0.8}	1	1085.62	0.84	2	1228.86	0.88
BMI-QSR _{0.9}	1	1083.12	0.85	2	1227.86	0.88
NN	3	606.17	0.81	4	520.72	0.86
NN-SR	3	1063.05	0.80	4	849.67	0.85
NN-QSR _{0.5}	3	901.14	0.82	4	772.99	0.85
NN-QSR _{0.6}	3	890.38	0.81	4	768.49	0.84
NN-QSR _{0.7}	3	884.44	0.81	4	755.39	0.85
BMI	3	824.04	0.89	4	728.95	0.90
BMI-SR	3	1371.47	0.89	4	1190.80	0.90
BMI-QSR _{0.6}	3	1150.63	0.89	4	1029.34	0.90
BMI-QSR _{0.7}	3	1129.05	0.89	4	1014.36	0.91
BMI-QSR _{0.8}	3	1100.16	0.89	4	994.94	0.90
BMI-QSR _{0.9}	3	1080.28	0.89	4	974.26	0.90
NN	5	490.47	0.88	6	437.09	0.90
NN-SR	5	777.17	0.88	6	682.67	0.89
NN-QSR _{0.5}	5	682.63	0.87	6	608.34	0.89
NN-QSR _{0.6}	5	670.30	0.87	6	601.73	0.89
NN-QSR _{0.7}	5	655.66	0.88	6	578.12	0.90
BMI	5	644.35	0.91	6	582.58	0.92
BMI-SR	5	1060.84	0.92	6	950.95	0.92
BMI-QSR _{0.6}	5	929.86	0.91	6	842.67	0.92
BMI-QSR _{0.7}	5	915.14	0.91	6	829.30	0.92
BMI-QSR _{0.8}	5	876.56	0.92	6	801.88	0.92
BMI-QSR _{0.9}	5	838.49	0.92	6	764.37	0.92

Table 3.8. Computational results obtained by the heuristic approaches

3.6 Conclusion

In the first part of this chapter we considered the dynamic version of the problem where knowledge of the entire sequence of actions is not assumed to be known *a priori*. The dynamic version of the Sport Event Filming can be treated as a Dynamic Vehicle Routing problem. We solved it by re-optimizing the position of the drones every time that a new action occurs.

In the second part of this chapter we proposed a mathematical formulation as multi-criteria optimization model by consider the minimization of the distance traveled, the maximization of customers and the minimization of the number of used UAVs. Concerning the mathematical optimization model we took into account of the dynamicity of the events by considering the concept of rolling-horizon. Furthermore, we proposed some and we compared the performance of heuristics proposed in the first part of the chapter, in terms of traveled distance, customer satisfaction and number of vehicles. In order to test and compare the heuristic with the mathematical formulation results, we considered a specific application scenario, that is a football match where the events were the game actions to be followed and our UAVs were equipped with cameras.

This work led to the writing and publication of the following works:

E. Natalizio, R. Surace, V. Loscrí, F. Guerriero, T. Melodia, “**Two Families of Algorithms to Film Sport Events with Flying Robots**”, in *10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2013)*, October 2013.

F. Guerriero, R. Surace, V. Loscrí, E. Natalizio, “**A Multi-objective Approach for Unmanned Aerial Vehicle Routing Problem with Soft Time Windows Constraints**”, in *Applied Mathematical Modelling, Volume 38, Issue 3, 1 February 2014, Pages 839-852*.

Multi-Objective Evolving Neural Network

Very recently, the idea of designing new communication devices, capable to adapt their operation roles in a self-organized fashion to rapidly face the changes within the working environment, has gained a very high attention from the wireless network research community [55],[56]; similarly, the availability of novel general purpose and powerful hardware platforms able to be dynamically reconfigured via software, has paved the way for new research directions in which it is possible to deploy extremely challenging communication scenarios.[57],[58].

Being inspired by this novel communication trend and, taking into account the unique features offered by the recent *Software Defined Radio (SDR)* paradigm, we considered the design of a self-adapting deployment strategy for a communication network in which several wireless devices, scattered all around in a specific area, can carry out a common task according to specific network requirements in terms of coverage or connectivity. In this context, aiming at handling very unlike communication scenarios, we firmly believe that, the SDR capabilities of future wireless devices can be effectively improved by coupling them with the potential offered by a wise *controlled mobility* strategy to exploit different reconfiguration and self-adaptation levels.

By following this first intuition, the work proposes a distributed *Neural/Genetic* algorithm to compute the final nodes positions and the more performing modulation schemes for each transmitter/receiver pair in order to guarantee an agreed QoS level. In particular, by considering a generic SDR architecture [59], the major advantages consisting into the ability of automatically selecting the more suitable modulation scheme to be used for an unknown received signal, can be effectively achieved. Thus, as a channel capacity varies, modulation scheme switching enables the baud rate to be increased or decreased in order to maximize the channel capacity usage. In addition, as demonstrated in our preliminary studies [61], SDR capabilities supported by a wireless node, coupled with the controlled mobility functionality, can improve the overall system performances in terms of connectivity. Therefore, such mobile SDR nodes turn out to be very useful for communication sce-

narios in which, the requirements on constrained QoS connectivity, are more stringent respect to the ones on the maximum coverage.

An example of such communication scenario could be the case of a *disaster area* where the communication between the survivors and the rescue teams has to guarantee a good quality level[60]; on the other side, a communication scenario mostly related to applications such as *pollution monitoring* or *fire detection* only need a high coverage degree that can be achieved by taking advantages just from the mobility of the nodes without using the SDR capabilities.

In summary, we made the following contributions in this work:

- we discussed the potential of mobile SDR communication devices in terms of both dynamical re-configuration and operation flexibility;
- we shown how the SDR capabilities supported by the wireless node, coupled with the controlled mobility functionality driven by a distributed Neural/Genetic, can improve the system performances in terms of connectivity by also guaranteeing an agreed QoS level;
- we designed and validated an optimization model in order to prove, in a mathematical way and on a very simple communication scenario, the goodness of the proposed Neural/Genetic algorithm before to conduct a more complex and intensive simulation analysis;
- we validated, throughout a self developed simulator based on a widely used open source framework for evolutionary design, the proposed strategy in different communication scenarios by varying both the amount of mobile and SDR nodes to measure the impact of a larger number of Mobile/SDR nodes on the overall system performances;
- we proposed an application of the neural/genetic algorithm on the video surveillance through ultra-low power sensors.

The rest of the work is organized as follows: Section 4.1 presents few recent research works on deployment techniques for wireless nodes and adaptive modulation schemes implemented via SDR. Section 4.2 describes which technologies and specific features the devices involved in the proposed communication scenario should support, by highlighting hardware and software capabilities. Section 4.3 presents the Neural/Genetic algorithm able to compute the best positioning for the wireless nodes to satisfy the constraints imposed by the specific communication scenario. Section 4.4 presents the proposed optimization model to describe and validate the system behaviour from a mathematical point of view. Section 4.5 discusses the obtained results and the goodness of the proposed approach in terms of extreme adaptability in different communication scenarios; in Section 4.6 we show an application of the algorithm on the video surveillance through ultra-low power sensors before the conclusions in Section 4.7.

4.1 Related Works

In this work we envision two main topics: 1) deployment techniques for different communication scenarios and 2) adaptive modulation schemes for nodes equipped with SDR capabilities. From the synergic combination and integration of *Neural* and *Genetic* approaches, we designed an algorithm able to compute the best position and the most suitable modulation scheme for each node involved into the communication path. According to these remarks, we first recall few contributions on positioning techniques for wireless nodes that have attracted much research attention becoming increasingly important in recent years. In [76] the authors try to outline the main criteria that should be considered while deploying wireless nodes in a sensor field. Fundamentally, they give an overview of multi-objective approaches by outlining the main assumptions and the formulation of this challenging problem. In [77], the author argues that the communication holes in wireless networks is the main problem causing inefficiency; thus it needs to be effectively addressed. More precisely, sensor nodes can be moved from an initial “unbalanced” state to a “balanced” state, where the number of communications holes is minimized. In [78] the authors distinguish placement approaches by considering deterministic and non-deterministic techniques. Often, non-deterministic placements are also named as random placements, while deterministic placements are referred as controlled placements and the authors refer to the two approaches by keeping this kind of assumption. The choice of how to deploy the sensors in a field, is often affected by the specific application, the type of sensors, the environment in which the sensors operate, etc. The possibility to control a node deployment can be extremely advantageous in terms of operational costs. Based on this last consideration, we figure out to equip wireless nodes with motion capabilities in order to make them able to move towards specific and more convenient locations by obtaining a dynamic changing of the topology/deployment. In this way, it could be add more control to the network and consequently outperform the operational costs. Generally, the survey we cited above, group the deployment techniques based on some specific network parameters such as coverage, lifetime, connectivity, or two or more of them together, but do neither consider at all the possibility for the nodes to autonomously move towards a specific position nor to adapt the modulation scheme to some specific requirements. In this work we consider the possibility to “change” the modulation scheme in order to guarantee a certain Bit Error Rate (BER) level, and in case this constraint can be satisfied by selecting more than one scheme, we select the most efficient in terms of energy consumption. For this reason, the contributions that compare modulation schemes selection to increase coverage and/or connectivity are considered also as related work. In [61] we present the preliminary idea to opportunistically select the more appropriate modulation scheme in order to achieve a certain degree of coverage and connectivity. The added value in respect of [61] is first of all in terms of a general benchmark definition, computed through the formula-

tion of the combined coverage and connectivity problems as a multi-objective optimization problem. Moreover, we provide more details in terms of results by validating the algorithm we propose in several communication scenarios. Starting from 2007, Stuckmann and Zimmermann [79] envisaged SDR technology as one of the four main objectives, to develop European technologies for systems beyond 3G. Specifically, the spectrum and resource management to make efficient the use of existing spectrum resources can be realized in a feasible and effective way through the application of the SDR concept. The importance of this kind of technology is shown in [80], where the authors propose to optimize the throughput of the network, working in different channel conditions, by considering an automatic modulation switching method to reconfigure the transceivers of SDR systems. We propose a similar approach but with a different purpose, namely a multi-objective algorithm where the goals are both the coverage and the connectivity, in addition the technique is also energy-aware because, where more solutions are feasible, the best one in terms of energy-consumption is selected. Finally, in [81], the authors analyse different modulation techniques in combination with SDR. In practice, they outline again the importance of this kind of technology for the future mobile communication systems.

4.2 Smart mobile devices supporting SDR

New powerful devices supporting SDR capabilities will be used in the next future to form a self-evolving wireless network in which several goals such as coverage increase, high data rate and connectivity will be achieved in diverse communication scenarios. For this reason, with the aim of considering a quite modern communication context, we studied the case in which both simple mobile or fixed sensor nodes equipped with a wireless IEEE 802.15.4/ZigBee compliant RF transceiver, are considered. Furthermore, we also took into account the presence of more complex mobile devices, with a high processing capability, able to dynamically change the modulation scheme between different transmitter/receiver pairs by using the SDR support as shown in Figure 4.1. In addition, we assume that these wireless devices are equipped with a GPS module coupled with a software application for position coordinates exchange to perform a specific positioning strategy as detailed in the next section.

By exploiting the new features of programmable SDR architectures, a flexible implementation of several modulation schemes (*i.e.* MFSK, MPSK, MQAM) can be realized in a simple and effective way. This flexibility turns into a great adaptivity to optimize different network performance indexes such as throughput, coverage and degree of connectivity of a wireless networks operating under varying channel conditions. In this context, the devices equipped with SDR functionalities can easily work as relay nodes in a multi-hop communication scenario by dynamically adapting different modulation schemes

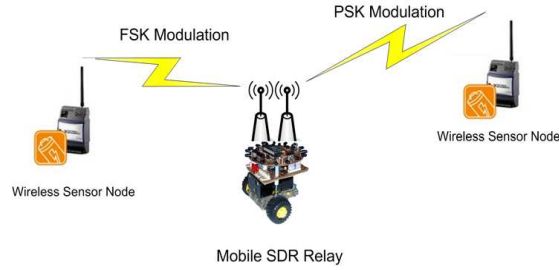


Fig. 4.1. Mobile relay devices supporting SDR capabilities

between the receiving and transmitting phases with the aim of optimizing network performances such as BER (Bit Error Rate), energy consumption and overall coverage.

It is well known that channel modulation has a relevant impact on the quality of the wireless link measured in terms of BER and on software/hardware complexity; furthermore, digital modulation/demodulation techniques need specific channel waveform coherence, coding/decoding and spreading/despreading of the radio spectrum [62]. Since the bit error probability is a function of the channel modulation, a radio channel with better quality has to be assigned to a larger number of bits and a higher order modulation, whereas a channel with poorer quality has to be assigned fewer bits or even no bit when the channel quality is too bad. For example, by working with three main digital modulation schemes (*i.e.* MFSK, MPSK, MQAM) having different modulation orders ($M=2,4,8,16$), it is possible to select and to use the most suitable combination by implementing it on-the-fly throughout SDR techniques. Moreover, to guarantee a certain BER value, the modulation schemes could be dynamically changed according to the channel quality experienced by the nodes and the distance variation between nodes due to the mobility.

In the last few years, the SDR paradigm is becoming more attractive and feasible thanks to the development of open-source software tool-kit such as GNU radio [64] and hardware devices such as Universal Software Radio Peripheral (USRP) [65]; therefore, several modulation/demodulation software blocks can be developed within the generic SDR architecture [63] for both transmitter and receiver (see Figure 4.2) allowing the design of new and more powerful devices well suited to support the dynamic modulation changing and adaptation strategies proposed in this work [58],[66].

In particular, it is worth to note that, with reference to the complexity of such software/hardware architecture, the Hybrid Radio Architecture (HYRA) proposed in [66] addresses the implementation of SDRs in the context of embedded systems by using reconfigurable hardware platforms with minimal additional resources.

Regarding the mobility features implementation, we would like to remark that this feature can be implemented by equipping a network device with two or four wheels and a servo motor controlled by Arduino-based modules. The estimated cost of implementing controlled mobility and reconfiguration capability in such platforms is reported in the Table 4.1. Even if this kind of software/hardware architectures seem quite expensive, the fast technological advancements will favour the reduction of estimated costs; thus, it is plausible that the mentioned embedded architectures will shortly be available at a much lower cost.

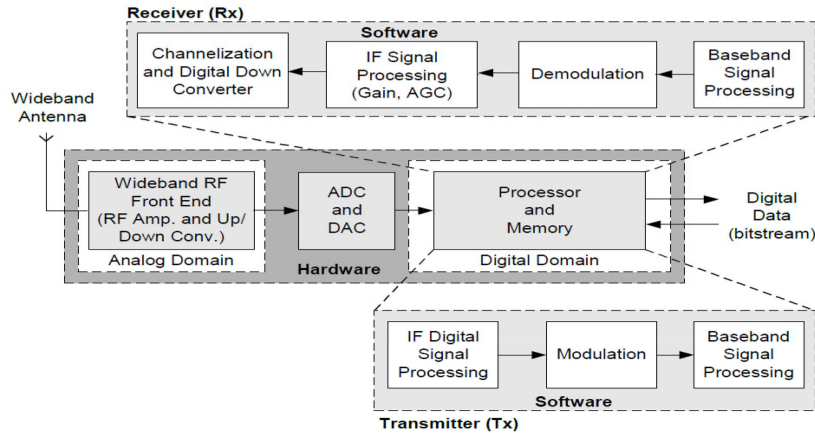


Fig. 4.2. SDR architecture for Relay Node [63].

Table 4.1. Costs to implement SDR devices.

Mobility Support	SDR Support
Wheels + Servo Motors + Arduino	USRP B200
250\$	800\$

4.3 Neural Network and Genetic Algorithm

The proposed work aims at computing, in a fully distributed fashion, the best positions for wireless devices belonging to a network placed in a given area, in order to satisfy some specific network requirements. In particular the

algorithmic scheme, to be performed by each node, consists of a neural network used to control and compute the next movement within the network area, and a genetic algorithm to perform a new solution that better fits with the desired objective function. Since the computation of the new nodes' position can be only based on local information, both the components of the proposed scheme are performed in a distributed way: a node executes the neural algorithm by knowing the positions of its neighborhood and the genetic algorithm manages its own genes without using global information.

Specifically, the two objectives considered in this work, namely, the coverage area and the number of sensor devices having a path toward a sink with a certain quality of service, are in contrast one to each other. Therefore we designed a wise strategy, based on the approach presented in [67], able to take into account both the requirements in a dynamic and re-configurable fashion.

4.3.1 The Neural Network

The neural network determines the movements of each wireless node; it is fully connected, recurrent and time-discrete. The neural network consists in input, output and hidden neurons. Inputs are subdivided as follows:

- 4 inputs to detect overlapping of sensing zone with neighborhood' sensing zone (1 for each direction);
- 4 inputs to detect missing of sink connection (1 for each direction);
- 1 to detect nodes in the same position.

The output is the new position. Each neuron "actives" a real-valued function and a time-varying real-valued connection with every other neuron of the network to map input (n-dim) in output (m-dim). 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 shown in Equation (4.1).

$$out_j(k) = F \left(\sum_{i \in N} w_{ij} \cdot out_i(k-1) + b_j \right) \quad (4.1)$$

where N is the set of neurons, w_{ij} is the weight of the connection between neuron i and neuron j and b_j is the bias of neuron j . Weights can produce both excitatory or inhibitory effect. The activation function F is the following linear threshold function:

$$F(x) = \begin{cases} -1.0 & \text{if } x \leq -1.0 \\ x & \text{if } -1.0 < x < 1.0 \\ 1.0 & \text{if } x \geq 1.0 \end{cases} \quad (4.2)$$

For each node the output of the neural network is given from the two output neurons and it consists of two real numbers that vary in the range

$[1, -1]$, as it is clear from (4.2) and Fig. 4.3. Based on these two values, the node chooses the action to do. Assuming a square field of $n \times n$ cells, the node can move in one of the four allowed directions or remain in the current cell.

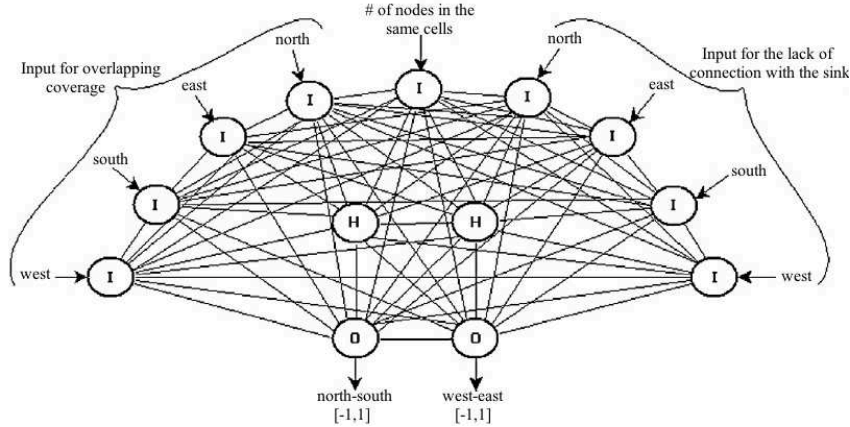


Fig. 4.3. Neural network architecture of one node

4.3.2 The Genetic Algorithm

A conventional and real-value Genetic Algorithm (GA) is used in the training phase of the Neural Algorithm. The genes are associated with the connections weights between each couples of neurons and the bias of each neuron. Through the typical operators of genetic approaches (*i.e.* crossover, mutation and selection), different weights to the neurons in the next generations will be assigned. Of course, the chromosome selected for the next generations is the one which has the best value of the fitness function. In our work, we consider a bi-objective functions in order to: 1) maximize the coverage; 2) maximize the number of nodes connected to the sink either in a direct fashion or through a multi-hop path. The fitness function can be written as follows:

$$fitness = \alpha * Coverage + \beta * QoS_{connectivity} \quad (4.3)$$

where

$$Coverage = \frac{covered_area}{whole_area} \quad (4.4)$$

and

$$QoS_{connectivity} = \frac{\#nodes_connected_to_the_sink}{\#total_nodes} \quad (4.5)$$

where α and β are weights that take into account the priority of the objective to be reached.

The term *#nodes_connected_to_the_sink* represents the number of nodes “connected” to the sink, namely nodes that are able to reach the sink directly or for which exists a multi-hop path [68] throughout links that are able to deliver data by guaranteeing a certain value of BER given as an input constraint. To the follow we show the pseudo-code of the Neural/Genetic Algorithm.

Algorithm 1 Neural/Genetic Algorithm

```

Random Deployment of Wireless Devices;
for all generation  $i$  do
  for all chromosome  $j$  do
    for all node  $n$  do
      while  $time < time_{MAX}$  do
        Compute the new position of  $n$  through the neural algorithm;
        Compute the modulations that  $n$  must use to reach either the other nodes and the
        sink by guaranteeing a specific QoS; (see Section 4.3.4);
        Among all the modulations satisfying the QoS choose those that require less energy
        (this only a possible choice, we could also consider the modulations that maximize
        the throughput, etc.) (see Section 4.3.5);
      end while
    end for
    Compute chromosome fitness  $j$ ;
  end for
  Consider the chromosome with the highest fitness value, apply genetic operators and then
  consider it as input for the next generation  $i+1$ ;
end for

```

4.3.3 Supported Connectivities and Communication Complexity

As already explained, each node has a set of possible modulations that can be “used” for data transmission by guaranteeing different connectivity levels. In this work, we figure out two different types of connectivities:

- *BasicConnectivity*: according to a specific propagation model, it is possible to compute the maximum distance at which a node n_1 is able to transmit, that is the transmitting radius. If a node n_2 is inside the area delimited by the circle with radius equal to the transmitting radius of the node n_1 , then n_1 and n_2 are connected. In this work we consider the propagation model as defined in [69]:

$$PL_{generic} = \left(\frac{4\pi d_0}{\lambda}\right)^2 \cdot \left(\frac{d}{d_0}\right)^\gamma + \chi \quad (4.6)$$

where d is the distance between the Tx and the Rx , γ is the path loss exponent, λ is the wavelength, χ is the shadowing effect value (neglectable) and d_0 is the critical distance.

- *QoSConnectivity*: each node n has a neighborhood. For every neighbor, the node n computes the BER value on the specific link by considering the different available modulations. The node n excludes all the modulations that do not respect the BER required as constraint in input. In practice, in this way the node n has a set of neighbors and every link from n to the

neighbor meets the QoS constraint in terms of BER computed according to the formulas shown in the next subsection.

For each iteration of the algorithm and for every node, both the *BasicConnectivity* and the *QoSConnectivity* are computed.

Since the proposed algorithm is based on local communication, a node only needs to know the position of its neighboring nodes to make a movement. According to [67], after each nodes movement, an update on the nodes position is broadcasted through a constant size message containing the node identifier (*Id*) and the node position (x,y), therefore the message size is in $O(1)$. As a consequence, considering a constant value for the number of time steps given as an input parameter, nodes will update their positions and broadcast their new information at each time step; this leads to a linear message sending complexity of $O(n)$ where n is the number of nodes within the network.

4.3.4 BER Computation

In order to compute the BER value related to each specific modulation scheme, we used the following relations coming from an asymptotic approximation [75]:

$$BER_{M-FSK} \approx 2^{k-2} \cdot \operatorname{erfc} \left(\sqrt{\frac{k E_b}{2 N_0}} \right) \quad (4.7)$$

$$BER_{M-PSK} \approx \frac{1}{k} \cdot \operatorname{erfc} \left(\sqrt{k \frac{E_b}{N_0} \sin^2 \left(\frac{\pi}{M} \right)} \right) \quad (4.8)$$

$$BER_{M-QAM} \approx 2 \frac{\sqrt{M} - 1}{\sqrt{M} k} \cdot \operatorname{erfc} \left(\sqrt{\frac{3k E_b}{2(M-1) N_0}} \right) \quad (4.9)$$

$$BER_{8-QAM} \approx \frac{5}{12} \cdot \operatorname{erfc} \left(\sqrt{\frac{1 E_b}{2 N_0}} \right) \quad (4.10)$$

In particular, the BER computation for the QAM modulation needs the use of two different formula according to the particular shape: formula (4.9) for squared modulations and formula (4.10) for non squared such as 8-QAM.

4.3.5 Transmitted Energy Computation

In order to save energy for the transmission always guaranteeing the required QoS level, it is necessary to choose the less power hungry modulation scheme within the set of the the most common modulation schemes available in real devices. To this aim, the energy spent per information bit [J] can be computed as follows [72]:

- For both MQAM and MPSK, by considering a signal bandwidth equal to $B[Hz]$ and, by assuming a sample time $T_s \approx 1/B$ [73], we can write:

$$E_{infBit} \approx \frac{(1 + \delta) \cdot SNR \cdot N_0 \cdot N_f \cdot G_d}{R} + \frac{P_c}{R \cdot B} + \frac{P_{tr} \cdot T_{tr}}{L} \quad (4.11)$$

where $\delta = \xi/\eta - 1$, ξ is the peak-to-average power ratio (PAPR) of the signal depending on the specific modulation, constellation size and shape¹ whereas η is the efficiency of PA drain chosen equal to 0.35 as typical value of class A power amplifiers [73]. P_{tr} and T_{tr} are the consumed power and the time spent in *transient* mode respectively whilst L is the total number of information bits. The $SNR = \frac{P_{rx} \cdot T_s}{N_0 \cdot N_f}$, where P_{rx} is the received signal power, $N_0/2$ is the power spectral density of the noise and N_f is the receiver noise figure. Moreover, by assuming a general path-loss model, the value of G_d can be computed according to the Equation (4.6) and it is equal to $G_d = \left(\frac{4\pi d_0}{\lambda}\right)^2 \cdot \left(\frac{d}{d_0}\right)^\gamma$. Finally, the term P_c represents the circuit power (*i.e.* 211 [mW] for both MQAM and MPSK) and the term R is the transmitting rate computed for each constellation by using the cutoff curves.

- For the MFSK modulation, by considering a noncoherent detection, the well known relation $M = 2T_S B$ [70] allows to derive $T_S = M/2B$, therefore the energy for the transmission will be equal to:

$$E_{infBit} = \frac{(1 + \delta) \cdot SNR \cdot N_0 \cdot N_f \cdot G_d}{R} + \frac{P_c \cdot M}{2 \cdot R \cdot B} + \frac{P_{tr} \cdot T_{tr}}{L} \quad (4.12)$$

where $\eta = 0.75$ is the typical value for class B or even greater (C, D, or E) power amplifiers [73], $\xi = 1$ according to [72] and $P_c = 165.3 [mW]$ for general MFSK modulation schemes.

4.3.6 Cut-off rate curves

In this work, we decided to use the relation between cut-off rate and pre-detection SNR to model the required power of the received signal; this choice is mainly motivated by the fact that the cut-off rate is considered as a meaningful measure of the effective maximum rate for convolutional coding with sequential decoding [70]. These relations can be calculated for MQAM and MPSK by the following formulas [71]:

$$R_0 = 2 \log_2(M) - \log_2 \left(\sum_{m=1}^M \sum_{i=1}^M C(x_m, x_i) \right)$$

¹ $\xi = 3 \cdot \frac{(\sqrt{M}-1)}{\sqrt{M}+1}$ for square constellations whereas M is the constellation size, while it assumes a value among those shown in Tab. 1 of the work [72] for cross-shaped constellations.

where $C(x_m, x_i)$ is the Chernoff bound on the pairwise error probability that for an AWGN channel having a Rician factor of $K = \infty$ we have:

$$C(x_m, x_i) = \exp\left(-\frac{1}{4}|d_{mi}|^2\right)$$

with $|d_{mi}|^2 = |x_m - x_i|^2/N_0$ and x_j is the j^{th} signal.

For noncoherent MFSK we have [70]:

$$R_0 = -\frac{1}{T_s} \cdot \log_2 \left\{ \frac{1}{M} + \left(1 - \frac{1}{M}\right) \cdot \exp\left(-\frac{\alpha^2}{2}\right) \left[\int_0^\infty x \cdot \exp\left(-\frac{x^2}{2}\right) \cdot \sqrt{I_0(\alpha x)} dx \right]^2 \right\} \quad (4.13)$$

where $\alpha^2/2 = SNR$ and $I_0(\alpha r)$ is the modified bessel function of the first kind. They are shown in Fig. 4.4.

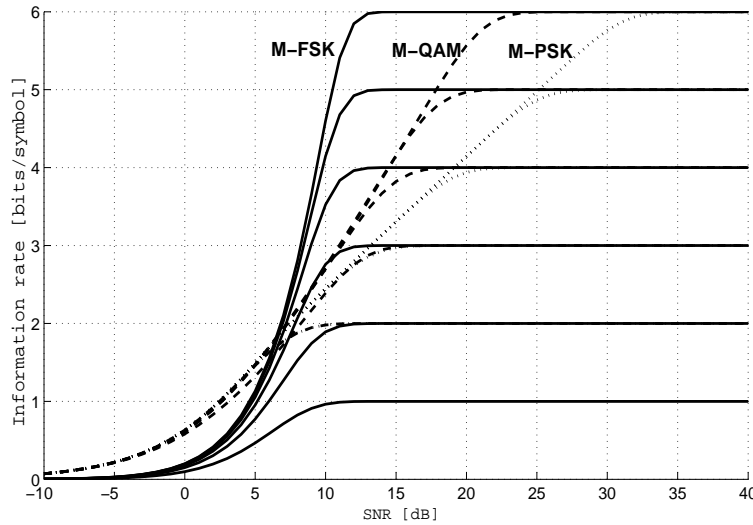


Fig. 4.4. Capacities and cutoff rates of MQAM, MPSK, and MFSK in AWGN channel as a function of predetection SNR. Each curve represents a different constellation size. The curves $1 \leq \log_2 M \leq 6$ for M-FSK, and $2 \leq \log_2 M \leq 6$ for M-QAM and M-PSK are shown, where M is the constellation size.

4.4 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 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_s : the discretization step; $sink$: a particular node that is located in a defined a-priori position; n : the total number of available nodes, including the $sink$; r : the sensing radius; $txradius$: the transmission 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 node k in the field;
- ϕ_{ijk} , $i = 1, \dots, \lceil h/d_s \rceil$, $j = 1, \dots, \lceil w/d_s \rceil$, $k = 1, \dots, n$ a binary variable that takes the value one if the location (i, j) is covered by node k , and zero otherwise;
- ϕ_{ijk}^+ , ϕ_{ijk}^- , $i = 1, \dots, \lceil h/d_s \rceil$, $j = 1, \dots, \lceil w/d_s \rceil$, $k = 1, \dots, n$ are support variables;
- δ_{ij} , $i = 1, \dots, \lceil h/d_s \rceil$, $j = 1, \dots, \lceil w/d_s \rceil$, a binary variable that takes the value one if the location (i, j) is covered by at least one node, and zero otherwise;
- γ_{ij} , $i = 1, \dots, n$, $j = 1, \dots, n$, a binary variable that takes the value one if the node i is linked to node j , and zero otherwise;
- x_{ij} , $i = 1, \dots, \lceil h/d_s \rceil$, $j = 1, \dots, \lceil w/d_s \rceil$, an integer flow variable that is used to represent the paths from each node to the $sink$;
- ψ_k^+ , and ψ_k^- $k = 1, \dots, n$ variables used to represent a relaxed version of the flow conservation constraints.

The considered problem can be mathematically stated as follows:

$$\max \alpha \times \left(\frac{\sum_{i=1}^{\lceil h/d_s \rceil} \sum_{j=1}^{\lceil w/d_s \rceil} \delta_{ij}}{h/d_s \times w/d_s} \right) - \beta \times \frac{\sum_{k=1}^n \psi_k^+ + \psi_k^-}{2 * (n - 1)} \quad (4.14)$$

$$r - |i - x_k| \geq M (\phi_{ijk}^+ - 1), \quad \forall i, j \text{ and } k = 1, \dots, n - 1 \quad (4.15)$$

$$r - |j - y_k| \geq M (\phi_{ijk}^- - 1), \quad \forall i, j \text{ and } k = 1, \dots, n - 1 \quad (4.16)$$

$$2 * \phi_{ijk} \leq \phi_{ijk}^+ + \phi_{ijk}^-, \quad \forall i, j \text{ and } k = 1, \dots, n - 1 \quad (4.17)$$

$$\delta_{ij} \leq \sum_{k=1}^{n-1} \phi_{ijk}, \quad \forall i, j \quad (4.18)$$

$$M \delta_{ij} \geq \sum_{k=1}^{n-1} \phi_{ijk}, \quad \forall i, j \quad (4.19)$$

$$|x_i - x_j| - tx_{radius} \leq M (1 - \gamma_{ij}^+), \quad \forall i, j = 1, \dots, n \quad (4.20)$$

$$|y_i - y_j| - tx_{radius} \leq M (1 - \gamma_{ij}^-), \quad \forall i, j = 1, \dots, n \quad (4.21)$$

$$2 * \gamma_{ij} \leq \gamma_{ij}^+ + \gamma_{ij}^-, \quad \forall i, j = 1, \dots, n \quad (4.22)$$

$$x_{ij} + x_{ji} \leq (n - 1) * \gamma_{ij}, \quad \forall i, j = 1, \dots, n \quad (4.23)$$

$$\sum_{j=1, j \neq i}^n x_{ij} - \sum_{j=1, j \neq i}^n x_{ji} + \psi_i^+ - \psi_i^- = 1, \quad \forall i = 1, \dots, n - 1 \quad (4.24)$$

$$\sum_{j=1, j \neq i}^n x_{sink j} - \sum_{j=1, j \neq i}^n x_{j sink} + \psi_{sink}^+ - \psi_{sink}^- = n - 1 \quad (4.25)$$

$$0 \leq x_k \leq [h/d_s], \quad 0 \leq y_k \leq [w/d_s], \quad \forall k \quad (4.26)$$

$$x_k, y_k \text{ integer}, \quad \forall k \quad (4.27)$$

$$\phi_{ijk}, \phi_{ijk}^+, \phi_{ijk}^- \text{ binary}, \quad \forall i, j, k \quad (4.28)$$

$$\delta_{ij} \text{ binary}, \quad \forall i, j \quad (4.29)$$

$$\gamma_{ij}, \gamma_{ij}^+, \gamma_{ij}^- \text{ binary}, \quad \forall i, j = 1, \dots, n \quad (4.30)$$

$$x_{ij} \geq 0, \text{ integer } \forall i, j = 1, \dots, n \quad (4.31)$$

$$\psi_i^+, \psi_i^- \geq 0, \text{ integer } \forall i = 1, \dots, n \quad (4.32)$$

The objective function in (4.14) maximizes the number of locations covered by at least one node and the number of nodes that reach the *sink*. Conditions (4.15) - (4.17) state that if the distance between the node k and the location (i, j) is lower than or equal to the sensing radius r than the variable ϕ_{ijk} takes the value one, otherwise it is set to zero. Constraints (4.18) and (4.19) 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 node and zero otherwise. Conditions (4.20) - (4.22) state that if the distance between the

node j and the node j is lower than or equal to the $txradius$ than the variable γ_{ij} takes the value one, otherwise it is set to zero. Constraints (4.23) ensure that a flow can be sent from node i to node j only if a link between the two nodes exists.

Constraints (4.24) and (4.25) represent a relaxed version of the flow conservation constraints, where the variables ψ_i^+ and ψ_i^- give a measure of the violation of these constraints.

Finally, conditions (4.26)-(4.32) represent the variable domain constraints.

The mathematical formulation reported above is an integer nonlinear programming model, where the nonlinearity is confined to the constraints (4.15) - (4.16) and (4.20) - (4.21).

To eliminate the terms with the absolute value, we introduce the additional constraints reported below:

$$d_{x_{ik}} \geq i - x_k \quad \forall i, k \quad (4.33)$$

$$d_{x_{ik}} \geq -i + x_k \quad \forall i, k \quad (4.34)$$

$$d_{y_{jk}} \geq j - y_k \quad \forall j, k \quad (4.35)$$

$$d_{y_{jk}} \geq -j + y_k \quad \forall j, k \quad (4.36)$$

$$d_{x_i x_j} \geq x_i - x_j \quad \forall i, j = 1, \dots, n \quad (4.37)$$

$$d_{x_i x_j} \geq -x_i + x_k \quad \forall i, j = 1, \dots, n \quad (4.38)$$

$$d_{y_i y_j} \geq y_i - y_j \quad \forall i, j = 1, \dots, n \quad (4.39)$$

$$d_{y_i y_j} \geq -y_i + y_j \quad \forall i, j = 1, \dots, n \quad (4.40)$$

Thus, constraints (4.15) - (4.16) and (4.20) - (4.21) are replaced by the following conditions:

$$r - d_{x_{ik}} \geq M (\phi_{ijk}^+ - 1), \quad \forall i, j, k \quad (4.41)$$

$$r - d_{y_{jk}} \geq M (\phi_{ijk}^- - 1), \quad \forall i, j, k \quad (4.42)$$

$$d_{x_i x_j} - txradius \leq M (1 - \gamma_{ij}^+), \quad \forall i, j = 1, \dots, n \quad (4.43)$$

$$d_{y_i y_j} - txradius \leq M (1 - \gamma_{ij}^-), \quad \forall i, j = 1, \dots, n \quad (4.44)$$

4.5 Validations, Simulations and Results

The proposed *Neural/Genetic* algorithm is evaluated by simulations using FREVO [82], an open source framework for evolutionary design. We took into account a 40×40 cells field, where 64 nodes are placed in a random way according to an uniform distribution. We considered one cell and one time step as discrete units of space and time, respectively. Also the sensing radius of the nodes is $r = 2$ [cells] 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). For the neural network, we use 9 input neurons, 2 hidden neurons and 2 output neurons. For the genetic algorithm, we use 300 chromosomes and 100 generations. All the results have been averaged over 10 different runs to respect a confidence interval of 95%.

In order to conduct a quite realistic analysis on the energy consumption, we chose to set transmitting power (12 dBm) and receiver sensitivity (-80 dBm) of our devices by referring to an off-the-shelf Bluetooth module made by *Roaming Networks* [74]. Table 4.2 summarizes all the other simulation parameters used.

Table 4.2. Values of the relevant parameters used for the simulations

Device Parameters	
Power spent in <i>transient</i> mode (P_{tr})	100 [mW]
Time spent in <i>transient</i> mode (T_{tr})	5 [μ s]
Wavelength (λ)	0.125 [m]
Information bits (L)	1000 [bits]
Receiver noise figure (N_f)	10
Bandwidth (B)	10 [kHz]
Scenario Parameters	
Path Loss Exponent (γ)	3.8
Critical distance (d_0)	1 [m]
PSD of the noise ($N_0/2$)	$10^{-15}/2$ [W/Hz]
Bit Error Rate ($BER_{threshold}$)	10^{-3}
Maximum number of time steps	100
Genetic Algorithm Parameters	
% of elite selection (e)	15%
% of mutation (mu)	45%
% of crossover (c)	30%
% of created offsprings (off_c)	5%
% of selecting an offspring (off_s)	5%

4.5.1 Validation of the Optimizazion Model

This first simulation campaign aims at validating the optimization model formulated in section 4.4. The presented results have been achieved by using LINGO 9.0 [52] for the mathematical model and they have been averaged over 100 runs with a confidence interval of 95%. For this simulation campaign we use a simple scenario with a 6×6 cells field, where $\{3; 4; 5\}$ nodes are placed in a random way according to an uniform distribution. Also the sensing radius of the nodes is $r = 1 [cell]$ and the transmission radius is $tx_{radius} = 1 [cell]$. The discretization step is $d_s = 1$ and the large positive number M is equal to 1000. Fig. 4.5 shows that the behaviour of the algorithm proposed is very close to the centralized optimum obtained through the mathematical model for each value of $alpha/beta$ within the Fitness function. In order to assess the behaviour of the proposed optimization model, test problems, characterized by a small field size and a limited number of nodes, have been considered. This choice is motivated by the fact that the intrinsic complexity of the model allows to solve, in a reasonable amount of time, only small size instances. For this reason, once we validated the satisfying accuracy of the optimization model respect to the proposed heuristic scheme, we conducted a more intensive simulation campaign to explore several configuration parameters with an higher number of nodes within a wider area.

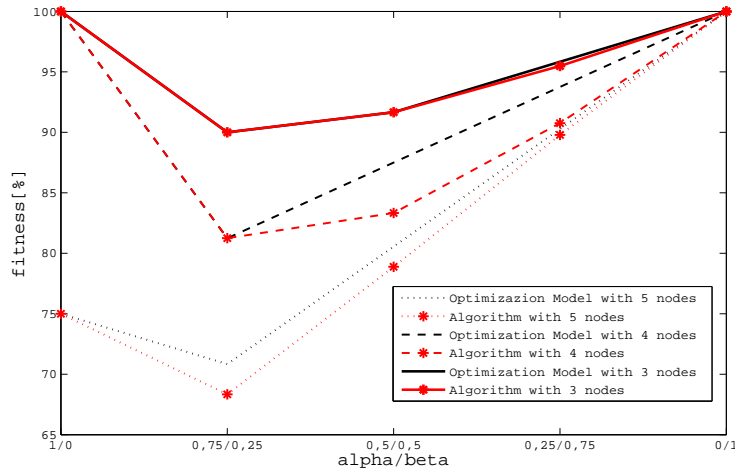


Fig. 4.5. Validation of the Neural/Genetic Algorithm: Heuristic vs. Optimization model.

4.5.2 Fixed nodes analysis supporting SDR

In this section we show the results obtained throughout the support of SDR capabilities; in this context the communication devices are all fixed but they can autonomously decide to use one of three different modulation schemes (MFSK, MPSK, MQAM) with three different symbol levels M (4, 8, 16) thus the set of possible choices is extended to nine. However, since the mobility support is out of the scope of this first reference simulation scenario, the Neural/Genetic algorithm described in section 4.3 cannot be executed every new generation and the result, in terms of more suitable modulation schemes, in agreement with the desired QoS, is always the same representing the reference benchmark point for the next analysis in which the mobility of the nodes allows to achieve better performances.

Figure 4.6, obtained throughout the implemented simulation framework, shows a clear example of a communication scenario in which the nodes are fixed but the use of SDR capabilities allows them to achieve different results in terms of $QoS_{connectivity}$. In particular, in figure 4.6.(a) the circles representing the nodes, are coloured in different ways according to the different supported modulation schemes (i.e., yellow for FSK, cyan for QAM and magenta for PSK). The square around the circle represents the communication ability of each node (i.e., blue if they can reach the sink node, red if they can communicate between each others without reaching the sink node). On the other site, the nodes displayed in figure 4.6.(b) are all coloured in blue because they can choose to use all the different modulation schemes according to the new features provided by the SDR technology. Thus, they can communicate with more neighbours respect to the previous scenario in order to reach the sink node by increasing the performance in terms of $QoS_{connectivity}$. On the contrary, the *Coverage* cannot take advantage from the SDR technology due to the lack of mobility support.

Table 4.3 summarizes the obtained results over 1000 simulation runs, also specifying the percentage of nodes that have chosen any specific modulation scheme and the average energy consumption for the transmission. It is worth to note that in the simulated scenario few modulation schemes such as 4-8-PSK have never been chosen due to the worst performance in terms of BER and to the higher energy consumption.

4.5.3 Mobile nodes analysis supporting SDR

In this section we show how controlled mobility can be efficiently exploited to reach better configurations both in terms of *Coverage* and $QoS_{connectivity}$. In Figures 4.7 we show results obtained when all nodes are equipped with motion capabilities but they are not able to select, in a dynamic fashion, the most suitable modulation. Specifically, all nodes will support only a specific modulation scheme in a random way, by keeping the percentage of nodes that choose a certain modulation equal for all the modulation schemes. In this specific case,

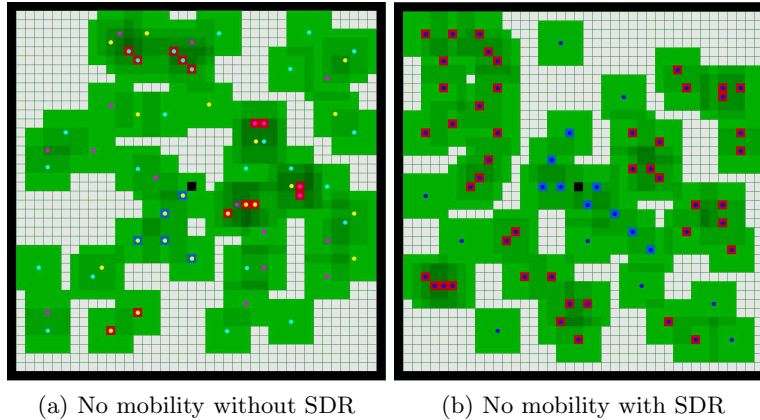


Fig. 4.6. Fixed nodes communication scenario: (a) No mobility without SDR, (b) No mobility with SDR.

Table 4.3. Fixed nodes analysis supporting or not supporting SDR capabilities

Output Parameters	Without SDR	With SDR
<i>Coverage</i>	61.57%	61.47%
$QoS_{connectivity}$	2.03%	12.20%
Energy per Information Bit	$2.4 \cdot 10^{-5} [J]$	$2.75 \cdot 10^{-5} [J]$
Nodes choosing 4-FSK	12.59%	0%
Nodes choosing 8-FSK	28.34%	38.57%
Nodes choosing 16-FSK	34.82%	43.92%
Nodes choosing 4-PSK	0%	0%
Nodes choosing 8-PSK	0%	0%
Nodes choosing 16-PSK	10.80%	10.94%
Nodes choosing 4-QAM	0%	6.57%
Nodes choosing 8-QAM	0%	0%
Nodes choosing 16-QAM	13.45%	0%

the 33.33% of nodes will support FSK or QAM or PSK modulation. Of course, in order to obtain reliable results we averaged them complying a confidential interval of 95%. In this scenario, all nodes will move in a distributed fashion towards novel positions computed through the neural network by considering a genetic approach during the training phase as explained in Section 4.3. It is worth to notice that in the configuration where nodes are not able to move, *Coverage* cannot be improved and nodes are only allowed to choose a better modulation in order to improve $QoS_{connectivity}$. When nodes are able to move, better configurations in terms of both *Coverage* and $QoS_{connectivity}$ are obtained. For α values ranging from 0.5 to 1, after 20 Generations the nodes are able to reach a percentage of coverage higher than 90%. Concern-

ing the $QoS_{connectivity}$ index, it is increased from 12.2% to 35% after ≈ 35 Generations by tuning the connectivity parameter (β) with higher values, 0.75 and 1. Unfortunately, if we observe the curves related to *Coverage* and $QoS_{connectivity}$ in a cross-way, we notice as controlled mobility is a valid tool to improve performance of the system, but coverage and connectivity are opposite goals, and then controlled mobility is able to generate configuration that “answer” in an effective way to the setting of α and β , but it is not able to handle the opposition of those two QoS requirements. In fact, in Figure 4.7 (a), when the value of α is chosen equal to 0.25, the network is not able to reach a degree of coverage higher than $\approx 24\%$, and this is the case (see Figure 4.7 (b)) where the connectivity value reaches $\approx 42\%$. By considering an additional freedom degree consisting into the possibility to set the most suitable modulation, the overall performance of the system are improved as shown in Figure 4.8 (a) and (b). A strange effect of the dynamic modulation setting occurs when α and β are both set equal to 0.5. In this case (see Figure 4.8 (a)), coverage reached is smaller than in the previous case, but it is worth to analyze this behavior in combination with the connectivity value. In fact, in Figure 4.8 (b), in correspondence of the same α and β values, we are able to obtain a connectivity degree higher than 93% after very few Generations. On the other hand, the system gives an answer matching the interest we express with the α value, since we set α equal to 0.5. When the α value is higher than 0.5, the system takes properly into account this setting and the coverage increases. From this analysis, we can argue that, by considering in a similar way the importance of both α and β parameters, the system will behave in a very effective fashion guaranteeing a very high level of $QoS_{connectivity}$ and a good degree of *Coverage*. These results are also confirmed by the *Fitness* curves shown in Figures 4.7 (c) and 4.8 (c) respectively. In fact, we can observe as *Fitness* improves by reaching very high values after a few number of generations when the weight associated with connectivity is the highest possible ($\alpha = 0$ and $\beta = 1$). In respect of the case in which nodes are only equipped with motion capabilities, SDR mobile nodes are able to react to the connectivity requests of the networks. Moreover, in all the studied cases we can notice an improvement of the *Fitness* except when coverage is considered as a kind of high priority (*i.e.* $\alpha = 0.75$ or $\alpha = 1$) making the *Fitness* trend similar to the case with no-SDR mobile nodes. As main conclusion of this simulation campaign, we can argue that, by correctly tuning the α and β weights of the *Fitness* function, the wireless network consisting of self-configuring SDR devices can dynamically react in order to face different communication scenarios by favoring, from time to time, the *Coverage*, the $QoS_{connectivity}$ or both.

4.5.4 Mobile nodes analysis with variable amount of SDR nodes

In this section we investigate the impact of the amount of SDR nodes on the overall network performances because, as detailed in section 4.2 and in table 4.1, the SDR nodes are still quite expensive devices; thus it is conve-

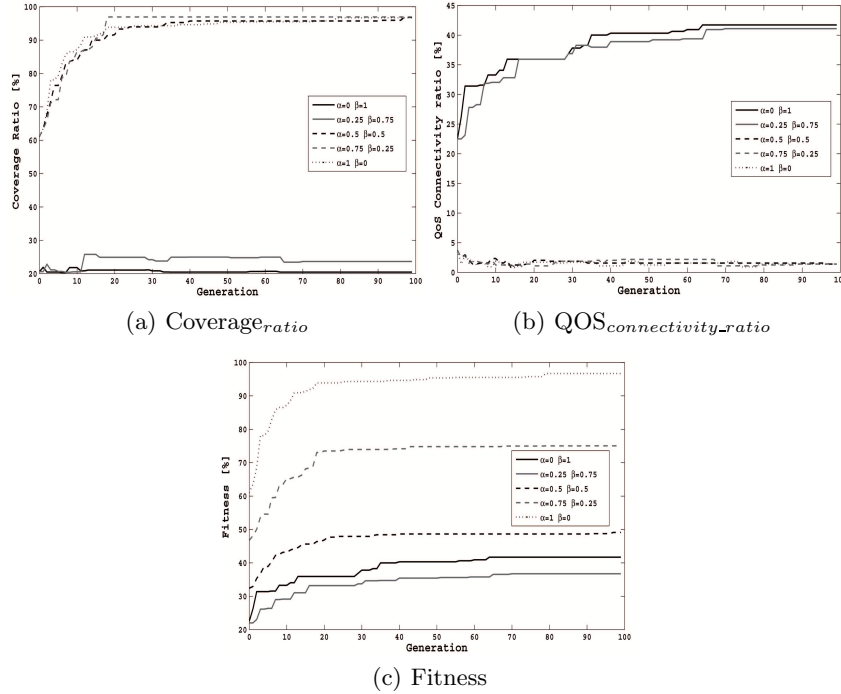


Fig. 4.7. Neural/Genetic algorithm supporting mobility without SDR capabilities

nient to reduce their number as much as possible. We tested the system with mobile nodes by choosing the same value for the parameters of the Fitness function (i.e. $\alpha = \beta = 0.5$) and varying the percentage of SDR nodes (i.e. 0%; 20%; 30%; 50%; 100%). The obtained results, shown in figure 4.9, demonstrate that even using a small amount of SDR nodes, it is possible to achieve good performances in terms of $QoS_{connectivity}$ (figure 4.9.b) but the *Coverage* turns out to be considerably reduced due to the fact that the nodes equipped with SDR capabilities work as attractors for the nodes without those features by greatly reducing the possibility to expand to cover larger areas.

4.5.5 Varying the percentage of mobile nodes

In this scenario we investigate the impact of high number of mobile SDR nodes on the overall network performances. This analysis is mainly motivated by the fact that the mobility capability has a quite expensive cost and the SDR equipment is still a bit expensive at the present day. According to these remarks, it makes sense to test how these capabilities impact on the performance of the network, both in terms of coverage and connectivity toward the sink; thus we consider only a portion of nodes equipped with SDR capabilities by varying

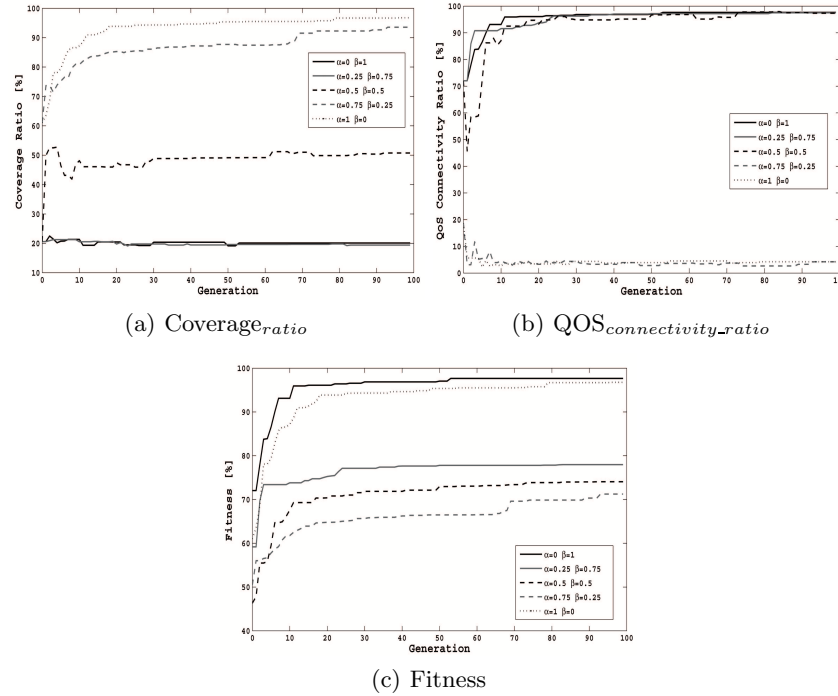


Fig. 4.8. Neural/Genetic algorithm supporting mobility and SDR capabilities

the number of nodes able to move toward “better” positions. In this way, we aim at dimensioning the right number of mobile nodes in order to save money without excessively reduce the network performances. With this goal in mind, we decided to test a network scenario in which only half of the nodes are provided with SDR functionalities and, among the standard nodes, a variable percentage are equipped with mobile capabilities (i.e. 40%; 60%; 80%; 100%). Just to give a numerical example, let us consider 64 nodes in the network field, 32 among them are equipped with SDR capabilities and are static whilst the number of mobile nodes, without SDR capabilities, varies as follows: 12, 19, 25 and 32.

The obtained results are shown in figure 4.10. As already explained, by using a certain percentage of SDR nodes it is possible to improve the network performances in terms of connectivity but at the expense of coverage; on the other side, by considering a more realistic network scenario in which not all the nodes can move, the effect of attraction mechanism, due to the SDR nodes, decreases thereby improving the performance in terms of coverage. However, if the percentage of standard mobile nodes is lower than 80%, we experienced a bad connection management toward the sink node; i.e. to reach the more

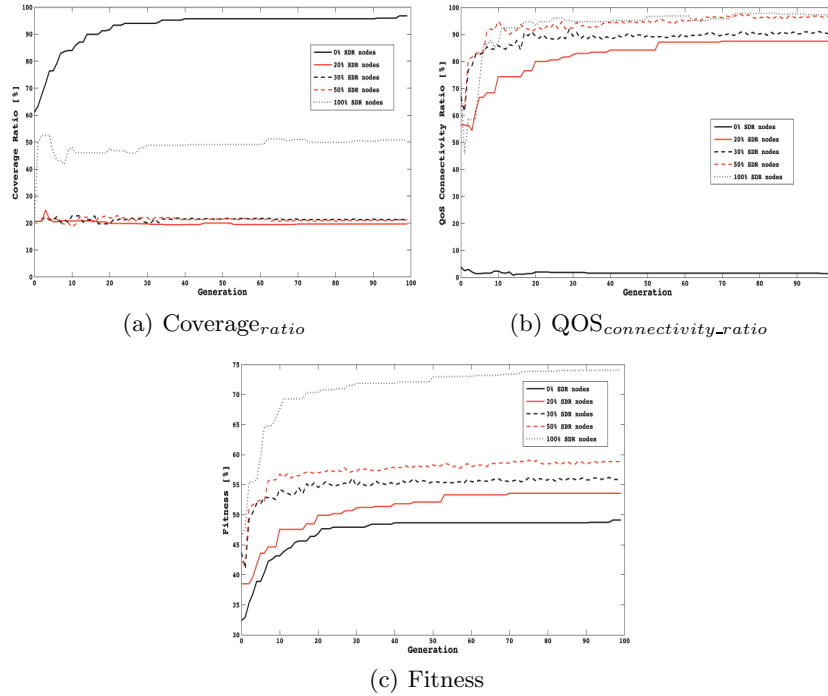


Fig. 4.9. Neural/Genetic algorithm supporting mobility and percentage of nodes with SDR capabilities

isolated fixed nodes, the SDR nodes prefer to ensure optimal coverage rather than communicate with the sink.

As main conclusion of this simulation campaign, we can argue that it is possible to decouple the effects due to both SDR and mobility features; in fact, in a mixed scenario in which only a portion of fixed nodes are equipped with SDR capabilities, a good $QoS_{connectivity}$ level can be guaranteed by increasing the number of mobile standard nodes.

4.6 Video Surveillance Applications based on Ultra-Low Power Sensor

In the last few years, a significant effort has been made in the context of wireless networks, by effectively exploiting their ability to monitor real-world phenomena [84], [86]. The applications involving wireless sensor networks are several and with different features, but one common factor of many applications, is the energy-constrained aspect of battery-powered devices. Normally, the wireless networks based on battery-powered devices, are mostly influenced

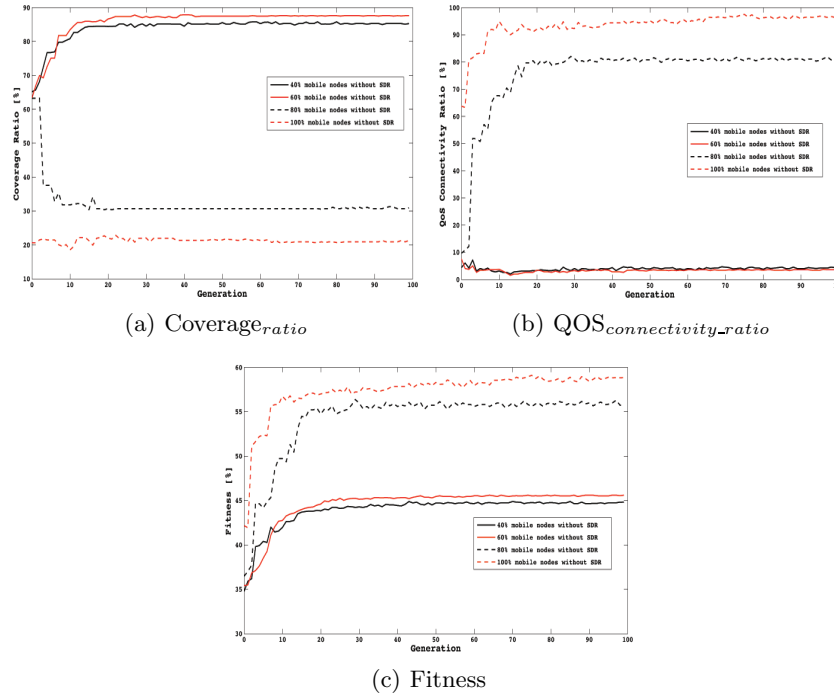


Fig. 4.10. Neural/Genetic algorithm supporting SDR capabilities and percentage of mobile nodes

by an effective and valid deployment of the nodes in the space. Deployment is concerned with setting up an operational heterogeneous wireless network in a real-world environment. Usually, the realization of an effective deployment is a labor-intensive and cumbersome task. Since energy is a limited and very precious resource, the extension of the lifetime of a battery-powered nodes network has to be addressed from different levels: 1) at the device level, by considering circuits with specific features; 2) at the network level, by implementing effective medium access solutions [91], routing protocols, deployments etc. In this work we try to devise a solution that combines an effective deployment of specific ultra low power wireless sensors nodes with LPSN sensors for monitoring objects' movements of specific areas. Specifically, LPSN nodes [87] are able to sense motion. Since VSC are energy-expensive nodes, it would be useful, for surveillance purpose [89], to wake-up video-camera [90] if and only if there is an interesting event that occurs (i.e. human being presence detected). In order to increase the “detectable” area, namely the zone where the events of interest can occur, and realize the maximum connected VSC nodes with the sensors (each VSC has to be connected to at least a LPSN sensor in order to be woken-up), we propose a neural/genetic approach. Neu-

ral/genetic approaches can be very effective for the solution of multi-objective problem as shown in [61]. This algorithm has the capability to consider in a synergistic way two “opposite” objectives. Usually, the greater is the area to be covered, the smaller is the number of VSC that are connected with at least a LPSN node. In order to take into account the two goals in a simultaneous way, two weight factors are introduced that give a kind of priority to the objectives. The rest of the work is organized as follows. Section 4.6.1 describes the problem we claim to resolve and the specific scenario considered. Section 4.4 presents the neural network exploited by each node to self-compute its best position. Section 4.6.2 describes the genetic algorithm used as training phase for the neural network. Section 4.6.3 presents the simulation results in different scenarios.

4.6.1 Reference Model and Problem Formulation

In this section, we will describe the specific characteristics of the LPSN sensors and the deployment problem. As described in [83] and [88], the Pyroelectric passive InfraRed (PIR) nodes, can be used as a trigger to wake-up a node from “sleep” mode to a power-hungry video capture mode. LPSN sensors are exploited in this specific context for event detection purpose [92], [93]. Specifically, they allow to sense motion and are able to detect if a human being is moving in or out of the sensor range. In the Figure 4.11, we show the architecture of LPSN nodes considered in this work. The hardware architecture is divided into three modules, powered by a single source:

- The sensor module which hosts a LPSN sensor, and the conditioning circuitry to give an analog and a digital output
- Microcontroller board, the controller module built around a TI MSP430, which includes the power harvester module and batteries
- The communication module consisting of a nanoWatt WUR circuit and an ADF7020 transceiver to send information and/or to wake-up the neighbor

More details about each module can be found in [83].

In this work the low-power LPSN sensor, that can sense the motion, is used to detect the presence through continuous low-power sampling. Once motion is detected, a signal is sent to turn on the video camera for higher resolution sensing of the event. The main motivation beyond the combined use of LPSN nodes and video camera, is that LPSN sensors exhibit significantly lower energy consumption. An example of how LPSN network can be inserted in an existing energy expensive sensor network is shown in Figure 4.12.

From Figure 4.12, we can notice as the existing network is not modified with the additional LPSN nodes and the LPSN sensor network is overlaid to the primary network. When a LPSN sensor detects an intruder, it broadcasts a message to its neighbourhood (the set of nodes that are in the cover range). The message is to wake-up the reachable nodes.

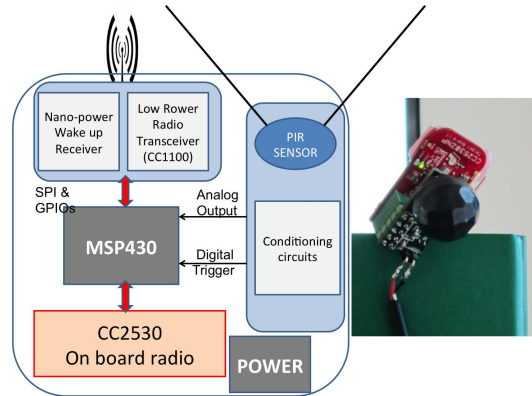


Fig. 4.11. Architecture and image of the implemented LPSN sensor node.

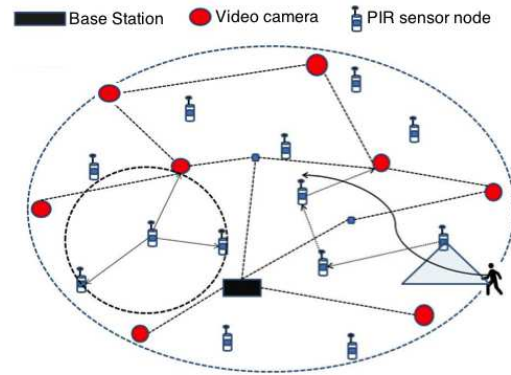


Fig. 4.12. Ultra-low power sensor network overlaid on an existing WSN.

Associated with the LPSN nodes are two different ranges: 1) transmission range to send broadcast messages (to the video camera) to wake-up them. We will refer to it as *wake-up radius*; 2) directional sensing range to detect events of interest. Whether a VSC is in the *wake-up radius* of a LPSN, this means that this node can be woken-up, since there is at least one LPSN able to wake-up it if some event occurs. We will refer to the number of video camera that can be woken-up with the term “connectivity” and our goal is to maximize it by the mean of a good deployment of the overlaid LPSN network. If the percentage of VSC that can be woken up is not equal to the maximum (100%), this means that some VSC can not react to some events and are isolated. On the other hand, we are also interested for each the maximum coverage, namely to maximize the areas covered through the sensing range of the LPSN nodes.

This means, that we ensure that all the events will be detected by at least a LPSN.

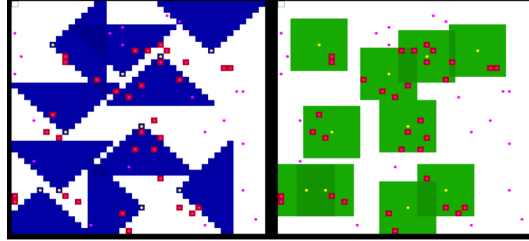


Fig. 4.13. (left) Example of coverage realized with 10 LPSN nodes (sensing radius = 8 meters); (right) connectivity example with 10 LPSN nodes (transmission range = 4 meters).

In the Figure 4.13, we show a deployment example realized with 10 LPSN nodes (yellow nodes in the picture) and 54 VSC (pink nodes). By setting a directional sensing radius (blue area) equal to 8 meters and an omnidirectional *wake-up radius* equal to 4 meters (green area), the LPSN sensors will be capable to wake-up a certain number of VSC (pink nodes with red squares) by covering an area (the covered area is the total blue area). The VSC that are out form the coverage of at least a LPSN (pink nodes), will be isolated and then cannot be woken-up. Follow this reasoning, we can argue that the objectives to maximize the number of video camera that can be woken up and the maximization of the detection areas are opposite. Based on these considerations, we formulate a neural/genetic approach, where we formulate the problem by considering two weight factors, that can be adapted to the specific requirements of the user (that would give priority to the connectivity, that is the number of VSC that can be woken up, or to the coverage).

In the next Section we will give the details for this approach.

4.6.2 Evolutionary algorithm for the neural network training phase

For the neural network, we use the same algorithm and parameters used in Section 4.3. In order to train the network, in a self-organizing perspective, unsupervised 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. The genetic algorithm is encoded with the neural network weights in a predefined manner where one gene in the chromosome represents one weight link. 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.

Since the goal of genetic algorithm is to find a population that permit to achieve the maximum value of a given fitness function, we need to relate the fitness function to a measure of coverage and time needed for coverage. To this scope the fitness function proposed for our scheme to make possible the evolution of the neural network is the following:

$$fitness_function = achieved_coverage - time \quad (4.45)$$

At each generation the fitness function (4.45) is evaluated and the new population encoding the weights's value 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. In this sense the fitness function is used as feedback for next generation. Notice that since time and space are discretized in this equation we're not summing seconds and meters but just counting how many cells are covered with current generation taking into account how many time steps are needed. Increasing the fitness function value by one unit in respect to previous generation means that evolution has led to cover one more cell or to cover the same number of cells but with one time step less. Table 4.4 shows the parameters used for the genetic algorithm.

Table 4.4. Parameters of genetic algorithms

Population size	100
Number of generation	100
Percentage of elite selection	15
Percentage of mutation	45
Percentage of crossover	30
Percentage of randomly created offsprings	5
Percentage of randomly selecting an offsprings from previous generation	5

4.6.3 Performance Evaluation

In order to compute effectively and adaptively a deployment of LPSN nodes that overlays the VSC network, we considered the neural/genetic technique described above. This algorithm is effective to compute the best deployment of the LPSN nodes, by responding to the specific requirements of the user, by taking into account the environment and the number of devices available. The simulation tool considered is FREVO [82]. The synergistic combination of the

neural network and the genetic algorithm is able to take into consideration different objective in a simultaneous way, by introducing two weights with value ranging from 0 to 1 (the sum of these two weight factors has to be equal to 1). As for instance, if we assign to the coverage weight 1, the weight connectivity will be 0, and this means that the coverage will be “prioritized”, and the connectivity will be not considered at all and viceversa. We tested the algorithm, by varying the weights between 0 and 1., the number of the LPSN nodes (10, 20, 30) and the wake-up radius (4, 8, 12 meters).

In the Figure 4.14(a) we show the results we obtain when the weight assigned to the *coverage* is 0 and the *connectivity* weight is 1.

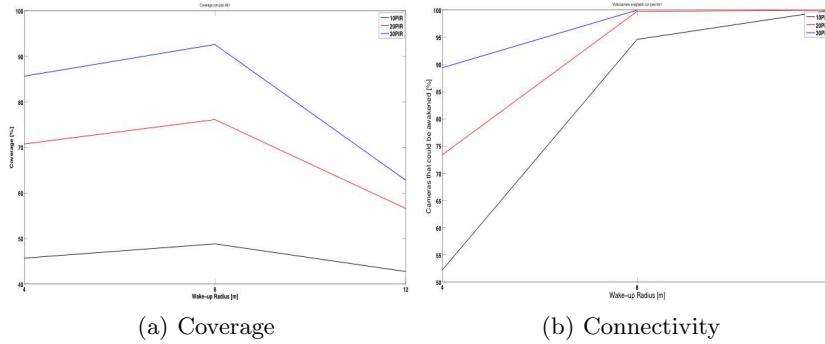


Fig. 4.14. Performance when the *coverage* is 0 and the *connectivity* is 1.

In Figure 4.14(b), we show the results concerning the connectivity, when the weight factor associated with the coverage is equal to 0 and the connectivity factor is 1.

In Figure 4.15(a) we show the results we obtain when the weight assigned to the coverage is 1 and the connectivity weight is 0.

In Figure 4.15(b), we show the results concerning the connectivity, when the weight factor associated with the coverage is equal to 1 and the connectivity factor is 0.

As we can remark from the Figures 4.14, when the user requires to the algorithm to prioritize the connectivity, the algorithm will deploy the nodes in a way to achieve the 100% of the VSC that can be woken up. In practice, the technique will try to cover a VSC with at least a LPSN node. This means that the coverage is not required to achieve the 100%. On the other hand, in the Figures 4.15, we can observe that the area can not be totally covered, by considering the specific number of LPSN nodes and the correspondent sensing radius to detect the events, but in any case, the coverage achieved is greater than in the previous case. On the other hand, the connectivity will achieve the maximum value by increasing the wake-up radius.

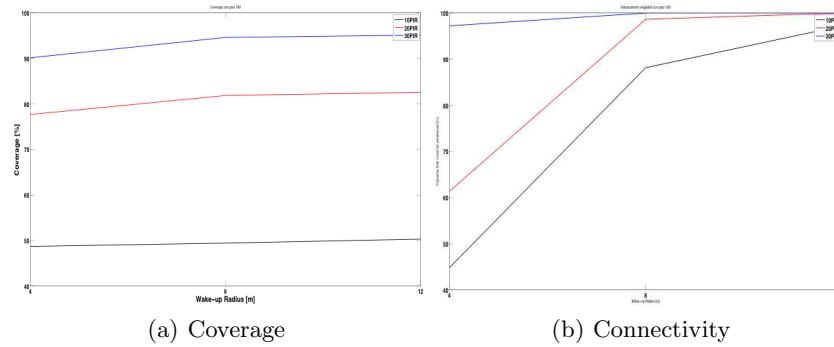


Fig. 4.15. Performance when the *coverage* is 1 and the *connectivity* is 0.

4.7 Conclusion

In the first part of the chapter we considered SDR mobile nodes able to move towards most suitable positions and to select the best modulation scheme in order to both improve the coverage within a specific area and the connectivity to a sink node. All nodes run an algorithm based on a totally distributed Neural/Genetic approach by using only local information. As main conclusion we can argue that the proposed strategy can handle the two opposite requirements (coverage and connectivity) in a dynamic fashion by wisely tuning the fitness function parameters; moreover, we demonstrated that, quite similar performances, can be achieved also using a reduced amount of devices equipped with expensive SDR and mobility features. In the second part of this chapter we have investigated on effective deployment of an overlaid network inserted over a network of Video Surveillance Camera. The deployment is dynamically computed through a neural/genetic approach, that allow to consider a synergistic combination of two parameters, named here “connectivity” and “coverage”, that normally are opposite. Through this approach we can ask the LPSN nodes to redeploy in order to improve the number of VSC that can be woken-up.

This work led to the writing and publication of the following works:

V. Loscrí, P. Pace, R. Surace, “**Multi-Objective Evolving Neural Network supporting SDR Modulations Management**”, in *24th IEEE International Symposium on Personal, Indoor, Mobile and Radio Communications (PIMRC 2013)*, September 2013.

V. Loscrí, M. Magno, R. Surace, “**Video Surveillance Applications based on Ultra-Low Power Sensor**”, in *1th International Workshop on Autonomous Monitoring and Networking (WAMN 2014)*, August 2014.

Decentralized Time-Synchronized Channel Swapping for WSN

Wireless sensor network (WSN) applications, such as visual sensor networks [94, 95, 96], mobile robots and drones [97, 98] and wireless capsule endoscopy [99], require:

- high bandwidth and energy efficiency to transmit large amounts of sensory data (images, video, acceleration and position data, etc.) with low latency and the smallest possible impact on each sensor's battery resources;
- spontaneous network reconfiguration and quick convergence to steady state when nodes join or leave the network;
- robustness to packet losses stemming from interference in the unlicensed 2.4 GHz band [100, 101] used by most WSN deployments.

Beyond these requirements, the recent thrust towards machine-to-machine (M2M) communications [102, 103] and the integration of WSNs with the generic internet infrastructure via 6LoWPAN support at the Network layer [104, 105] call for the development of *infrastructure-less*, peer-to-peer, communication protocols at the Medium Access Control (MAC) layer.

The concept of channel hopping has gained acceptance as a good solution for WSN MAC layer coordination, with the time synchronized channel hopping (TSCH) [106] protocol now being part of the IEEE802.15.4e-2012 standard [107]. Within the context of WSNs, channel hopping enables nodes to hop between the 16 channels of the 2400-2483.5 MHz band. This is performed such that transmitters and receivers are evenly spread across channels, so nodes are not constantly in a channel with excessive interference.

Fig. 5.1 depicts an example of the TSCH protocol [106], where an arbitrary topology is formed between 14 nodes (depicted at the bottom) [108]. Each node reserves timeslots within the slotframe interval (horizontal axis of the top part) and within the 16 channels of IEEE802.15.4 (vertical axis of the top part). Unoccupied slots appear in white. As the slotframe interval of Fig. 5.1 repeats periodically, all nodes transmit and listen in different channels, thus avoiding concentrated interference.

However, the TSCH slotframe has a rigid (pre-defined) structure (see Fig. 5.1) and filling up the available slots follows a rather complex advertising request and acknowledgment (RQ/ACK) process on a coordination channel. This channel is prone to interference and occasional self-inflicted collisions when the nodes are set to advertise slot reservations very aggressively. Conversely, if slot advertising is not aggressive and nodes leave the network, their slots may remain unoccupied for long periods until another advertisement RQ/ACK process reassigns them to other nodes. This limits the bandwidth usage per channel, as illustrated by the large number of unoccupied slots of Fig. 5.1. Finally, TSCH cannot be considered as an infrastructure-less protocol, as (i) a coordinator node is required in order to maintain global time synchronization via beacon message broadcasts at slotframe boundaries [101, 106, 107, 108]; and (ii) a dedicated coordination channel must be available for the advertisement RQ/ACK process.

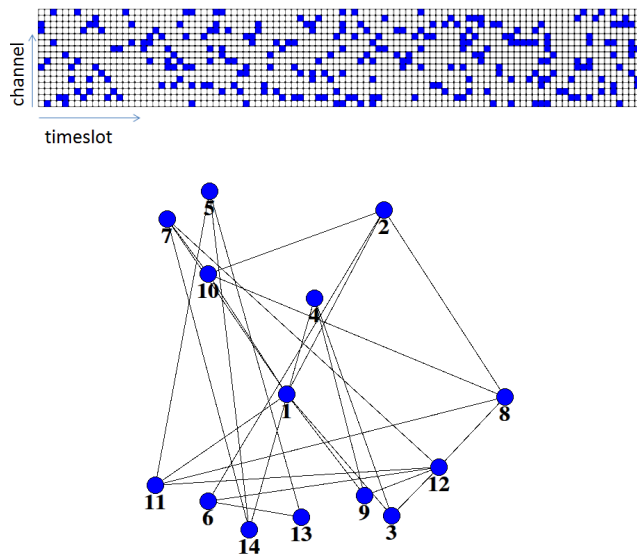


Fig. 5.1. TSCH example of 14 nodes derived by the 6tisch simulator: (top) slotframe structure with 101 timeslots and 16 channels of IEEE802.15.4 (blue indicates used slots, white indicates unused slots); (bottom) corresponding connectivity mesh.

5.1 Related Work

Several approaches propose the use of multichannel MAC layer coordination in a number of ways. The first category comprises schemes that assign channels to nodes in a *static* manner in order to balance them across the available

channels of IEEE802.15.4 and maximize bandwidth utilization [109, 110, 111]. Such solutions, however, try to minimize the rate of channel hopping, as this affects network stability and tends to decrease the achieved transmission rate per node. As such, they achieve reduced node connectivity and are prone to persistent interference in any of the utilized channels.

The second category comprises protocols based on *dynamic* coordination of node channel hopping throughout the lifetime of the WSN. Hwang *et al.* [112] proposed a low-energy, receiver-driven, channel hopping scheme for WSNs that does not require global time synchronization. Instead, each sender predicts the wake-up time of each receiver encountered, which is shown to minimize idle listening at the cost of significantly reduced bandwidth efficiency. Tang *et al.* [113] proposed the Efficient Multichannel MAC (EM-MAC), a multichannel protocol based on receiver initiated predictive wake-up. In EM-MAC, nodes select the channel for communication by following pseudo-random scheduling, as in the predictive wake-up approach. EM-MAC is shown to be highly resilient to interference and jamming with similar energy characteristics to predictive wake-up MAC, albeit at the cost of substantially reduced duty cycling and low bandwidth utilization.

Alternative approaches for multichannel coordination and hopping utilize a control (or “coordination”) channel, where nodes negotiate the channel to use for data transmission. Representative examples include Y-MAC [114], A-MAC [115], MMAC [116], CAM-MAC [117], MuChMAC [118] and the TSCH option of IEEE802.15.4e-2012.

Y-MAC [114] uses a hybrid of contention and scheduling mechanism for access control. Scheduling the receiver wake-up times helps to minimise idle listening and overhearing. A base station or “sync” node sends timing packets on a control channel to start the network. These packets also serve as a way to provide for time synchronization. Transmitting nodes first compete during a contention-based broadcast period to transmit to the receiver during a slot in the unicast period. Contention-based backoff methods are used within the broadcast section to schedule slots in the unicast section. Periodic control frames, containing time synchronisation data, are also sent so as to keep the broadcast–unicast frame structure aligned.

A-MAC [115] establishes an optimal timeout value for each node to periodically wake up so as to send and receive packets. In conjunction with a rate estimation scheme, A-MAC is shown to decrease energy consumption compared to previous approaches.

Mobile adaptive MAC (MMAC) [116] is designed as a protocol suitable for mobile nodes by using dynamic frame times to allow nodes to send data without long waiting periods, i.e., before the network topology changes. Transmission is contention-free, with frame time calculated as a function of node mobility. Nodes are required to know their location and movement, which is a disadvantage since nodes must invest large amounts of energy to establish this information (e.g., via GPS data).

Cooperative asynchronous multi-channel MAC (CAM-MAC) [117] uses *co-operation* at the center of the protocol design. Nodes creating new connections are advised by neighbors as to which channels would cause the least disruption when all neighbors cannot be heard. This is done by the transmission of probe packets announcing a node's intention to establish a new connection. Neighboring nodes are then allowed to provide feedback to the probe in the form of an invalid response, indicating the connection is not in their interest. If the probe goes uncontested, then the connection is established, otherwise the probe fails and the sending node creates a new session elsewhere.

MuChMAC [118] is a low-overhead dynamic multi-channel MAC for wireless sensor networks. The protocol was designed to be general-purpose, suitable for a wide range of traffic rates. Energy efficiency is achieved by very low duty cycle (few percent) and collisions are minimized by utilising subslots within each TDMA timeslot determined by the node ID. This gives performance similar to other multichannel TDMA protocols under high traffic load, while performance under light traffic is similar to single-channel protocols.

Finally, time-synchronized channel hopping (TSCH) [106] is a frequency hopping reservation based protocol that uses timeslots to make a frame. TSCH uses a control channel where nodes can advertise their free timeslots. Receiving nodes can compare their free slots with the advertised slots and schedule a connection by answering the advertisement during the preferred slot. The protocol was designed for networks that may contain mobile nodes and stemmed from previous work on the TSMP protocol and the wirelessHART standardization [119, 120]. In comparison to previously-proposed protocols, TSCH strikes a good balance between bandwidth utilization, energy consumption and node connectivity. Therefore, it was adopted as an optional mode within the IEEE802.15.4e-2012 standard [121, 107]. Beyond its standardized version, TSCH is currently developed via the open-source openWSN effort and the related 6tisch simulator for an associated IETF RFC [108] and therefore, can be considered as the definite benchmark for multichannel MAC protocols.

From the previous description, it is evident that the bandwidth and reliability of the control node (or channel) can become significant obstacles to the efficacy of multichannel protocols for decentralized processing and communications applications arising in many mobile and ad-hoc WSN infrastructures [94, 95, 97, 98], especially under strong interference conditions. These issues are expected to become even more pronounced within infrastructure-less deployments envisaged for M2M and 6LoWPAN WSNs [102, 103, 104, 105]. For these reasons, on-going efforts towards a *decentralized* TSCH mechanism [106] that does not rely on a single coordinator node or coordination channel employ distributed Aloha-based scheduling for the advertisement channel and a gossip mechanism for the propagation and response to advertisement information. However, such mechanisms: *(i)* are still based on the time and energy-consuming RQ/ACK mechanism; *(ii)* have a rigid slotframe structure (Fig. 5.1) and *(iii)* require an independent manner for global time synchronization (e.g., via a separate GPS unit [106]).

Overall, with the aim to improve channel hopping protocols for WSNs, three key issues can be identified: *(i)* avoiding the dependence on a coordination channel and/or coordinator node and converting the time-frequency coordination into a *truly decentralized* framework; *(ii)* providing a decentralized approach for time synchronization in the WSN; *(iii)* making node synchronization and timeslot assignment dynamic under varying interference conditions and densities of nodes per channel.

5.1.1 Novelty, Contributions and work Organization

This work addresses these issues based on the concept of *pulse coupled oscillators* (PCOs) [122, 123]. Specifically, we propose a novel *decentralized time-synchronized channel swapping* (DT-SCS) framework, in which nodes randomly join a channel and achieve PCO-based coordination via the periodic transmission of *beacon* packets at the MAC layer.

For channels with an equal number of nodes, DT-SCS converges to synchronized beacon packet transmission at the MAC layer in a completely uncoordinated manner. Furthermore, it allows for arbitrary pairwise swaps between nodes in neighboring channels with minimal effort and without disrupting the WSN operation. Finally, due to the inherent adaptation of PCO mechanisms to the effects of nodes joining and leaving the process, our proposal is robust to interference and the effects of node churn during WSN reconfiguration.

While the use of PCOs as a means for decentralized synchronization or desynchronization is well-established for single-channel distributed coordination [122, 123], this is the first approach to simultaneously deploy *both concepts* for *decentralized* time-synchronized transmissions and channel swapping.

Our detailed contributions are summarized below:

- We propose the DT-SCS protocol that marries the key benefits of: *(i)* a self-managed, decentralized, collision-free, time-division multiple access (TDMA) schedule produced by PCOs with negative coupling within each channel (a.k.a. DESYNC [122]); *(ii)* PCOs with positive coupling (a.k.a. SYNC [123]) *across multiple channels* to provide for spontaneous alignment of nodes' timeslots, which in turn allows for node pairwise channel swapping without losing WSN stability or bandwidth efficiency; *(iii)* elastic (rather than rigid) time synchronization and spontaneous adaptation of the available transmission slots across all channels via the SYNC/DESYNC coupling coefficients.
- We prove that DT-SCS converges to a balanced steady state and estimate its expected connectivity and energy consumption.
- We present detailed simulation and experimental results demonstrating the efficacy of the proposed protocol for distributed multichannel coordination in WSNs.

- Finally, we carry out detailed comparisons between DT-SCS and TSCH in terms of: convergence delay, bandwidth efficiency, robustness to interference and the existence of hidden terminals, and energy consumption.

Regarding the remainder of the work, 5.2 describes DT-SCS based on positive and negative PCO coupling. 5.3 analyzes the proposed protocol in terms of stability, connectivity and energy consumption. 5.4 provides simulation results, whereas Section 5.5 provides validation results on a WSN comprising TelosB motes with the Contiki operating system. Finally, Section 5.6 concludes the work.

5.2 PCO-based Sync/Desync for DT-SCS

5.2.1 Introduction to the Basic Concept

Consider a WSN consisting of W nodes randomly distributed in C channels [see the left part of Fig. 5.2(a)], with each node transmitting short *beacon* packets periodically every T seconds. The proposed DT-SCS mechanism balances the number of nodes per channel and adjusts the transmission time of each node’s beacon packets to reach an evenly-distributed timeslot allocation within each channel. Specifically, the nodes in each channel perform PCO-based desynchronization (i.e., they are “DESYNC” nodes) and elect a single “SYNC” node to provide for cross-channel synchronization. Within each period, the SYNC node of each channel listens for the SYNC beacon message in the next channel¹ and adjusts the transmission time of its own beacon packet in its own channel using PCO-based synchronization [123]. SYNC nodes will also move to the next channel if they detect that less nodes are present there. In this way, the WSN can converge to the steady state with $W_c = \frac{W}{C}$ nodes per channel². The beacon packet transmission flow between DT-SCS nodes is schematically illustrated in the right part of Fig. 5.2(a).

Once the system reaches the steady state, SYNC or DESYNC nodes in adjacent channels can swap channels and timeslots in pairs using a simple RQ/ACK scheme. Fig. 5.2(b) highlights the short interval between two consecutive beacon packet transmissions (stemming from two different nodes in a channel), during which RQ/ACK packet transmissions for channel swaps take place. If nodes join or leave the network, all remaining nodes adjust their beacon packet timings spontaneously, in order to converge to a new steady state. As shown in Fig. 5.2(a), the key aspect of DT-SCS is the spontaneous

¹ with cyclic behavior between Channels 1 and 16 of IEEE 802.15.4, i.e., the SYNC node of Channel 16 listens for the SYNC beacon message of Channel 1

² For simplicity, we assume that W is divisible by C . However, when this is not the case the scheme balances the number of nodes to $W_c \in \{\lfloor \frac{W}{C} \rfloor, \lceil \frac{W}{C} \rceil\}$ nodes per channel.

convergence of the WSN from a random state to a multichannel time synchronized beaconing, without the need for a coordinator node or a coordinating channel.

Once convergence to steady state is achieved, the only overhead in the proposed DT-SCS protocol stems from handling swap requests as well as beacon packet broadcasts. Both, however, are very short packets (less than ten bytes), which makes the overhead minimal compared to the payload packet transmission and reception.

The loss of beacon packets and timing errors due to interference cause node beacon times to *waver*, i.e., nodes send beacon messages at incorrect times. As such, all nodes receiving these messages are similarly affected. If left untreated, this wavering may propagate through the network until all nodes are affected and the network is no longer considered converged. To combat this, we consider the notion of *coupling* between nodes, introduced by PCOs [123, 124]: instead of a DESYNC node jumping directly to the midpoint of its beacon neighbors, the node *slides* towards the mid point with coupling factor α ($0 < \alpha < 1$); similarly, a SYNC node gradually adjusts its beaconing time by coupling factor β ($0 < \beta < 1$) to align with the beacon of the SYNC node in the next channel. Using PCOs with appropriate coupling factors ensures that any noise and instability in beacon timings is attenuated and does not propagate uncontrollably throughout all nodes and channels of DT-SCS.

5.2.2 Coupling via Sync and Desync

Next, we briefly describe the synchronization and desynchronization primitives that form the basis of the proposed DT-SCS. Both of these primitives are algorithms for revising the beacon packet broadcast time of a node in a WSN based on the broadcast times of beacon packets from other nodes within a certain time interval. Consider W_c nodes being present in channel c , with $c \in \{1, \dots, C\}$, and the total WSN nodes given by $W = \sum_{c=1}^C W_c$. Each node joins the network by broadcasting an initial beacon packet randomly in channel c at a time between $[0, T)$ seconds. Each node repeats the transmission of its beacon packet upon the completion of its cycle, namely, every T seconds. For each node, the fraction of the way through its cycle at a given time $t \in [0, T)$ is denoted as the node's *phase* [123, 124], $\varphi \in [0, 1)$.

As shown in Fig. 5.3, we can imagine the beacon packet transmission times as beads moving clockwise on a ring with period $T = 1$ s [122]. When the phase of a node becomes one (namely, the bead reaches the top of the ring in Fig. 5.3), a beacon packet is broadcast and its phase is reset to zero. Each node keeps the phase of received beacon packets and updates its own beacon phase $\varphi_{\text{curr}}^{(k-1)}$ to $\varphi_{\text{curr}}^{(k)}$ based on the utilized reactive listening primitive. Thus, superscript (k) indicates the k th *phase-update iteration*.

We remark that, for the SYNC/DESYNC algorithms, it is immaterial which physical sensor node is linked to which beacon broadcast, as the phase update process is solely dependent on the received beacon packet times [122, 123,

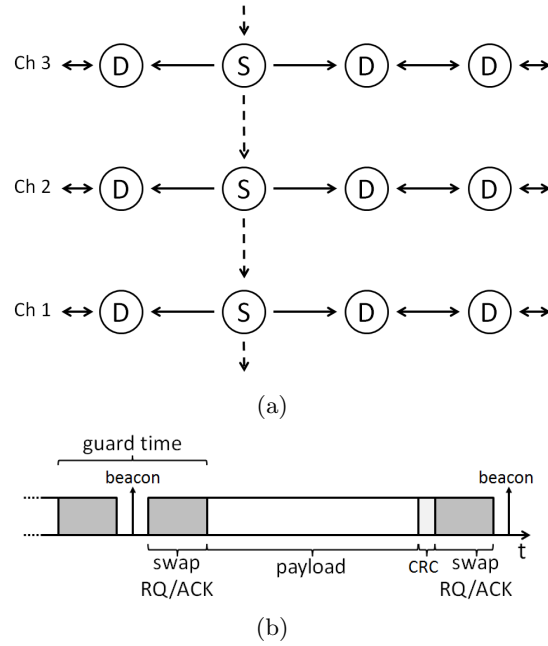


Fig. 5.2. (a, left) Initial random state of $W = 12$ node in $C = 3$ channels; (a, right) DT-SCS converged state with $W_c = 4$ nodes per channel, showing the intra-channel desynchronization (solid lines) and inter-channel synchronization (dashed lines) between DESYNC (D) and SYNC (S) nodes, respectively. Arrows indicate the intended recipient of each beacon packet transmission. (b) The grey slots indicate the short transmitting/listening intervals where nodes can request and acknowledge swaps.

[124, 125, 126]. For this reason, we shall be explicitly discussing *beacon packet transmission events* and not the physical nodes that broadcast them.

Desync Phase Update

During desynchronization in channel c , each node’s beacon phase is updated once within each period T . As shown in Fig. 5.3(a), the phase of a node “curr” is updated based on the phases of received “prev” and “next” beacon messages, originating from nodes that transmitted their beacon before and after node “curr”, respectively. Specifically, upon receiving the next beacon packet, the phase of node “curr” moves towards the middle of the interval between the phases of “prev” and “next” beacon messages. The k th phase update of DESYNC is expressed by³ [122, 125]

³ Since (5.1) is applied when the next beacon packet is received, we have that $\varphi_{\text{next}}^{(k-1)} = 0$ [see Fig. 5.3(a)]. However, we include $\varphi_{\text{next}}^{(k-1)}$ in (5.1) to clarify that

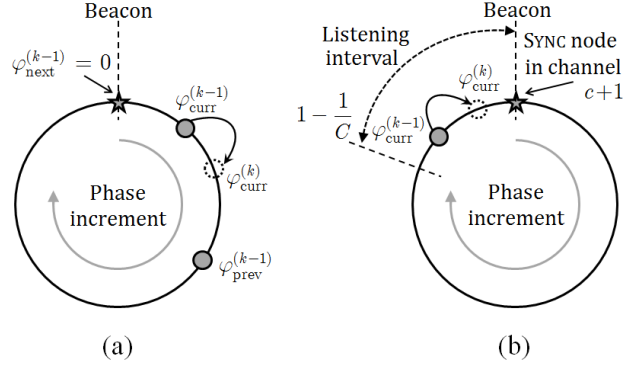


Fig. 5.3. (a) A DESYNC node performs its k th phase update when the next DESYNC or SYNC beacon packet is received in channel c . (b) A SYNC node performs its k th phase update when a SYNC beacon packet is received in channel $c + 1$ while the phase of the current beacon broadcast is within its listening interval.

$$\varphi_{\text{curr}}^{(k)} = (1 - \alpha) \varphi_{\text{curr}}^{(k-1)} + \frac{\alpha}{2} \left(\varphi_{\text{prev}}^{(k-1)} + \varphi_{\text{next}}^{(k-1)} \right) \pmod{1}, \quad (5.1)$$

with $\alpha \in (0, 1)$ the DESYNC phase-coupling constant controlling the speed of the phase adaptation and $\text{expr} \pmod{1}$ being the modulo-1 of expression $\text{expr} \in \mathbb{R}$. Previous work [122, 125] showed that the reactive listening primitive of (5.1) disperses all beacon packet broadcasts in each channel $c \in \{1, \dots, C\}$ at intervals of $\frac{T}{W_c}$. This leads to fair TDMA scheduling in channel c in steady state (SS). After k_{ss} iterations of (5.1), all beacon packets in channel c are periodic and the phase updates lead to convergence to SS, expressed by

$$\left| \varphi_{\text{curr}}^{(k_{\text{ss}})} - \varphi_{\text{curr}}^{(k_{\text{ss}}-1)} \right| \leq b_{\text{thres}}, \quad (5.2)$$

with b_{thres} the preset convergence threshold, typically $b_{\text{thres}} \in [0.001, 0.100]$. In steady state, each node in channel c transmits data packets for $T \left(\frac{1}{W_c} - b_{\text{thres}} \right) - t_{\text{swap}}$ seconds immediately following its beacon packet broadcast, where t_{swap} denotes the duration of the guard time per node in channel c . Hence, the maximum number of nodes supported under collision-free TDMA per channel c is less than $\left\lfloor \frac{1}{b_{\text{thres}}} \right\rfloor$.

Proposed Sync Phase Update

PCO-based synchronization with positive coupling [123] updates each SYNC node's beacon phase according to received beacon packets (from other SYNC

the operation of DESYNC depends on both the previous and next beacon packet phase.

nodes) that are within a *listening interval* of duration $\frac{T}{C}$ s [see Fig. 5.3(b)]. Under the proposed DT-SCS protocol, the phase of each SYNC beacon in channel c changes after a SYNC beacon packet is received in channel $c + 1$ within the listening interval. Hence, the k th phase update of PCO-based synchronization [123] is performed at $\varphi_{\text{curr}}^{(k-1)}T$ s after the node's last beacon packet transmission, $1 - \frac{1}{C} < \varphi_{\text{curr}}^{(k-1)} < 1$,

$$\varphi_{\text{curr}}^{(k)} = (1 + \beta) \varphi_{\text{curr}}^{(k-1)} - \beta \left(1 - \frac{1}{C}\right) \pmod{1}, \quad (5.3)$$

with $\beta \in (0, 1)$ the phase-coupling constant controlling the speed of the phase adaptation. Any beacon packets transmitted outside the listening interval $(1 - \frac{1}{C}, 1)$ are ignored with respect to the SYNC phase update. However, they are still processed to extract useful information, such as the total number of nodes in the current channel (see Section 5.2.3). After \tilde{k}_{ss} phase updates, (5.3) converges to coordinated SYNC beacon packet broadcasts at intervals of $(1 \pm \tilde{b}_{\text{thres}}) \times T$ seconds [123]. Similar to the DESYNC case, \tilde{b}_{thres} is used⁴ to detect convergence to SS under (5.2).

5.2.3 Protocol Description

It is evident that in a WSN comprising W nodes and utilizing coupling via SYNC or DESYNC within C channels ($C > 1$), the maximum attainable throughput per node will be achieved when the number of nodes is balanced across all channels, i.e., when $W_c = \frac{W}{C}$ nodes are present within each channel c , $1 \leq c \leq C$ [111]. Fig. 5.4 presents the basic stages of the proposed DT-SCS protocol, explained in the following subsections.

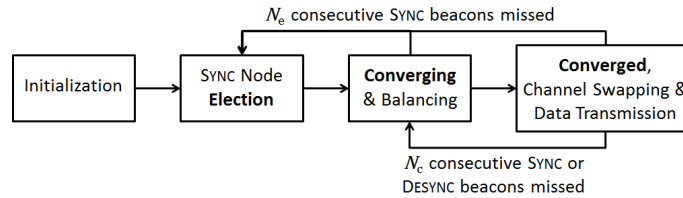


Fig. 5.4. Block diagram of the operational modes of DT-SCS. The values of N_e and N_c are set via experimentation with varying packet loss. Data transmission and channel swapping takes place only during the *Converged* mode.

⁴ Generally, the thresholds b_{thres} and \tilde{b}_{thres} for the respective cases of DESYNC and SYNC can have different values. However, in our implementation, we consider $b_{\text{thres}} = \tilde{b}_{\text{thres}}$.

Node Initialization and Beacon Packet Contents

When initialized, each node joins a channel c ($1 \leq c \leq C$) randomly as a DESYNC node. Initially, nodes have their receivers enabled for the entire beaconing period and send their beacon messages according to the DESYNC rules.

Each beacon packet transmitted by a node in channel c contains:

1. the type of beacon packet (SYNC or DESYNC);
2. the node unique identity number (node id);
3. the node id of the SYNC node in channel c (NULL if none);
4. the number of unique nodes heard in channel c , W_c ;
5. the number of unique nodes heard in channel $c + 1$, W_{c+1} ;
6. the current mode that the node perceives channel c to be in (Fig. 5.4): *Election* mode, *Converging* mode, or *Converged* mode.

Each node can independently establish the information of parts 3 and 4 by listening on channel c . The information for part 5 is obtained when the SYNC node in channel c listens to the beacon packet from the SYNC node in channel $c+1$. Alternatively, this information can also be obtained when DESYNC nodes in channel c listen for an acknowledgment of a swap request and overhear a DESYNC beacon in channel $c+1$. Finally, the information in part 6 is acquired as described in the following two subsections.

Election Mode

Election of a SYNC node is initiated in each channel c when N_e consecutive periods have passed without receiving a SYNC beacon packet, or when nodes observe that all other nodes report the SYNC node id as NULL. The value of N_e can be set high enough to avoid reelecting a SYNC node just because SYNC beacon packets were lost due to interference. In our experiments, we found that $N_e = 10$ provided for virtually no reelections when a SYNC node is already present in each channel c , while allowing for fast network response when a SYNC node actually leaves the channel.

Once the nodes in channel c ($1 \leq c \leq C$) go to *Election* mode, they report this in part 6 of their beacon packets. Each node then randomly generates an 8-bit number, $r \in [0, 255]$, and transmits it in part 3 of its beacon packet. After one complete period, the node with the highest number is elected to become the SYNC node for this channel. In the unlikely case where the highest number is sent by more than one node, the node with the highest node id (part 2) is elected. All nodes confirm the selection in the subsequent periods by setting their SYNC node id (i.e, part 3) to the node id they have elected. Because beacon packets may occasionally be lost, there may be some sporadic cases where nodes may not unanimously agree to the same elected SYNC node. In such cases, nodes rectify their election according to the majority decision. Once all nodes set the SYNC id field to the same value, the *Election* mode

(i.e., part 6 in the beacon message) changes to either *Converging* or *Converged* mode. This process ensures that (up to) one SYNC node is present per channel.

Converging Mode via Node Balancing across all C Channels

When nodes are in the *Converging* mode, no channel swapping takes place. However, in order to balance nodes within the available C channels, SYNC nodes can decide to switch to the next channel if less nodes are present therein, as described next.

During the *Converging* mode, all nodes apply the DESYNC and SYNC processes of Section 5.2.2. Once a SYNC node is elected in channel c , all DESYNC nodes receive information about the number of nodes present in the next channel, i.e., W_{c+1} (part 5 of beacon packet contents). If

$$W_c - W_{c+1} - 1 \geq 0 \quad (5.4)$$

and $c < C$, then the SYNC node of channel c switches to channel $c+1$ and joins as DESYNC node, while a new SYNC node is elected in channel c . Importantly, nodes in the highest channel, C , can switch to channel 1, i.e., perform “cyclic” switching from highest to lowest channel, if

$$W_C - W_1 - 2 \geq 0. \quad (5.5)$$

This prevents a race condition where nodes would be constantly moving across channels.

Via the new SYNC node of channel c , all nodes remaining in channel c will observe that W_{c+1} increased by one. Furthermore, after N_c consecutive misses of the beacon of the node id that switched, W_c is decreased by one, i.e., the node is confirmed as having departed channel c . The requirement of N_c consecutive misses before assuming that the node has left channel c avoids erroneously decreasing W_c due to a burst of packet losses in the network caused by transient external interference in channel c .

The above process will lead to nodes moving from lower to higher channels, thereby enabling the WSN to converge to a balanced number of nodes across all C channels. That is, if W is divisible by C , then $\forall c : W_c = \frac{W}{C}$ after balancing. An example of this case is illustrated in Fig. 5.2(a). Moreover, an illustrative example of balancing when W is not divisible by C is shown in Fig. 5.5. In the latter case, with respect to channel swapping (discussed in the following subsection), DESYNC nodes can apply channel swapping only between channel 1 and channel 2, and between channel 3 and channel 4. However, SYNC nodes can still swap places between all four channels as their beacon packet transmissions remain synchronous. Finally, the example of Fig. 5.5 demonstrates that, without the special condition of (5.5) for channel C , nodes would be cyclically switching during *Converging* mode.

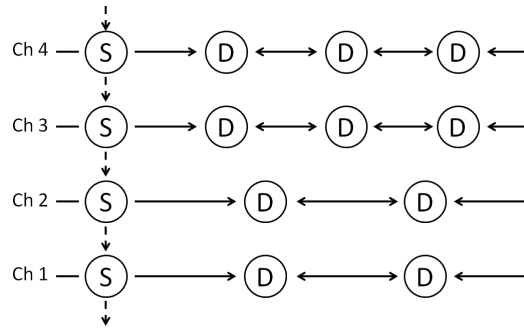


Fig. 5.5. Example of balancing under DT-SCS for a network of $W = 14$ nodes in $C = 4$ channels.

Converged Mode, Channel Swapping and Data Transmission

Once nodes are in *Converging* mode and their SYNC or DESYNC beacon packets fall within the convergence threshold, i.e., (5.2) holds, they move to *Converged* mode. They can thus begin data transmission following a short guard time interval after their beacon packet broadcast. The duration of their transmission lasts until another short guard time interval prior to the subsequent node beacon packet broadcast, as shown in Fig. 5.2(b). In *Converged* mode, nodes transmit data, send or acknowledge swap requests and swap channels, as explained next.

Firstly, the guard time is used to allow for beacon variability due to SYNC or DESYNC beacon time adaptation via (5.1) and (5.3). This adaptation may cause beacon time fluctuations due to packet losses (the range of these fluctuations is controlled via the coupling parameters α and β).

Secondly, in *Converged* mode all DESYNC nodes of each channel c can opt to transmit swap requests in the next channel, or acknowledge swap requests from a node of channel $c - 1$, if, and only if, $W_{c+1} = W_c$, or $W_c = W_{c-1}$ (respectively, with $W_{C+1} \equiv W_1$ and $W_0 \equiv W_C$). If a swap is acknowledged, the corresponding SYNC or DESYNC nodes swap channels in the subsequent period and remain in the new channel until another swap RQ/ACK event. Because the swap acknowledgment may not always be received by the requesting node, sporadic cases may occur where the node requesting the swap does not actually swap channels. To overcome this, every node that received a swap request transmits its first beacon packet towards the end of the guard time after performing the channel swap. This enables the node to detect that its swap-requesting “partner” is not sending its beacon in its old channel and has carried out the swap. If, however, the swap partner did not carry out the swap, then the node returns to its original channel, resumes beaconing therein and requests a new swap.

Via the channel swapping mechanism, DT-SCS ensures each node can attempt to swap channels whenever: (i) the application requires; (ii) a node

wants to reach nodes not present in their original channel; *(iii)* excessive interference (i.e., high packet loss) is observed in a channel.

Channel swapping should not be confused with channel switching to balance the total WSN nodes in C channels during the *Converging* mode, which is done without requests and acknowledgments.

Finally, as depicted in Fig. 5.4, nodes in the *Converged* mode may move back to *Converging* mode if N_c consecutive beacon packets are not received from any SYNC or DESYNC node. Nodes move to *Election* mode if N_e consecutive SYNC beacon packets are not received. In both of these modes, no data transmission or channel swapping takes place.

5.2.4 Discussion

Maximization of Swapping Possibilities

In DT-SCS, the SYNC node per channel c provides a single fixed phase reference to all the nodes in the channel. This allows for accurate phase swapping whilst maintaining convergence, without the need for *(i)* a control channel, *(ii)* globally synchronized clocks amongst nodes, or *(iii)* a network coordinator. Although all the SYNC nodes are always aligned once the *Converged* mode is reached, only channels with the same numbers of nodes, contain aligned DESYNC nodes. According to DT-SCS, channels with equal numbers of nodes are clustered together, with an ascending number of nodes per channel (see Fig. 5.5). This is an important feature of the algorithm, as it does not permit channels with unequal node counts to be interspersed. As nodes can only swap with their counterpart in neighboring channels (a node in channel c may only swap with the concurrently-firing node in channel $c + 1$), ensuring that channels with equal numbers of nodes are grouped together greatly improves connectivity in the network.

Local Coordination and Timing Elasticity

Swapping between SYNC nodes does not cause a reelection as both channels will still have a single SYNC node after the swap. One can perceive each SYNC node as a local channel “coordinator” with influence limited to channel c (its transmission channel) and channel $c + 1$ (the channel it listens to). This approach alleviates the need for a global network coordinator or coordination channel. Even if a SYNC node leaves the WSN, the remaining DESYNC nodes in the channel will elect a new node to take its place and the WSN will return to the *Converged* mode.

As mentioned in Section 5.2.1, losing beacon packets due to interference leads to beacon packet transmission at incorrect times. To absorb any transient oscillations of beacon packet times while at the same time maintain fast convergence, the values of coupling parameters α and β can be adjusted per node. Values for α and β close to unity allow for quicker convergence and

better suit channels with low interference, whilst values close to zero provide for more stable operation in noisy channels at the cost of longer convergence times and slower reaction times. There are well-known ways to select optimum values for α and β for a given application environment [122, 123], which are not elaborated in this work in detail.

5.3 Protocol Analysis

5.3.1 Balancing and Stability

As described in Section 5.2.3, during the *Converging* mode of the proposed DT-SCS protocol, SYNC nodes can decide to switch to the next channel if they detect less nodes present therein. We prove below that this mechanism leads to a balanced number of nodes per channel as illustrated in Figs. 5.2(a) and 5.5.

Proposition 1 *The proposed node balancing mechanism converges to $W_c \in \{\lceil \frac{W}{C} \rceil, \lfloor \frac{W}{C} \rfloor\}$ nodes within each channel c , $1 \leq c \leq C$.*

Proof. During the *Converging* mode, a SYNC node may switch from channel c to $c+1$, or from channel $c-1$ to c . A SYNC node switch occurring simultaneously between channels $c-1 \rightarrow c$ and $c \rightarrow c+1$ at the k th period can be expressed stochastically for the number of nodes in channel c ($1 < c < C$) by

$$\begin{aligned} \overline{W}_c^{(k+1)} = \overline{W}_c^{(k)} - u \left[\overline{W}_c^{(k)} - \overline{W}_{c+1}^{(k)} - 1 \right] p_{c+1}^{(k)} \overline{W}_c^{(k)} \\ + u \left[\overline{W}_{c-1}^{(k)} - \overline{W}_c^{(k)} - 1 \right] p_c^{(k)} \overline{W}_{c-1}^{(k)}, \end{aligned} \quad (5.6)$$

while for channel C ,

$$\begin{aligned} \overline{W}_C^{(k+1)} = \overline{W}_C^{(k)} - u \left[\overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2 \right] p_1^{(k)} \overline{W}_C^{(k)} \\ + u \left[\overline{W}_{C-1}^{(k)} - \overline{W}_C^{(k)} - 1 \right] p_C^{(k)} \overline{W}_{C-1}^{(k)} \end{aligned} \quad (5.7)$$

and for channel 1,

$$\begin{aligned} \overline{W}_1^{(k+1)} = \overline{W}_1^{(k)} - u \left[\overline{W}_1^{(k)} - \overline{W}_2^{(k)} - 1 \right] p_2^{(k)} \overline{W}_1^{(k)} \\ + u \left[\overline{W}_C^{(k)} - \overline{W}_1^{(k)} - 2 \right] p_1^{(k)} \overline{W}_C^{(k)}, \end{aligned} \quad (5.8)$$

where $u[\cdot]$ is the unit-step function, $\overline{W}_{c-1}^{(k)}$, $\overline{W}_c^{(k)}$ and $\overline{W}_{c+1}^{(k)}$ are the expected numbers of nodes at channels $c-1$, c and $c+1$ during the k th period, and $p_c^{(k)}$ is the probability the SYNC node will successfully switch to channel c during the k period. We remark that $p_c^{(k)}$ is smaller than unity since (typically) only

a single (SYNC) node will switch channels—or, in the case of interference or no SYNC node elected in the channel, no node will manage to switch.

For every channel $c \in [1, C]$ the transition system formed by (5.6) is written in matrix form as

$$\bar{\mathbf{w}}^{(k+1)} = \mathbf{G}^{(k)} \bar{\mathbf{w}}^{(k)} \quad (5.9)$$

with

$$\bar{\mathbf{w}}^{(k+1)} = \left[\bar{W}_1^{(k+1)} \ \bar{W}_2^{(k+1)} \ \dots \ \bar{W}_{C-1}^{(k+1)} \ \bar{W}_C^{(k+1)} \right]^T, \quad (5.10)$$

$$\bar{\mathbf{w}}^{(k)} = \left[\bar{W}_1^{(k)} \ \bar{W}_2^{(k)} \ \dots \ \bar{W}_{C-1}^{(k)} \ \bar{W}_C^{(k)} \right]^T \quad (5.11)$$

and

$$\mathbf{G}^{(k)} = \begin{bmatrix} 1 - g_1^{(k)} & 0 & 0 & \dots & g_C^{(k)} \\ g_1^{(k)} & 1 - g_2^{(k)} & 0 & \dots & 0 \\ 0 & g_2^{(k)} & \ddots & \dots & 0 \\ \vdots & \vdots & \ddots & 1 - g_{C-1}^{(k)} & 0 \\ 0 & 0 & 0 & g_{C-1}^{(k)} & 1 - g_C^{(k)} \end{bmatrix} \quad (5.12)$$

where

$$\forall c < C : g_c^{(k)} = u \left[\bar{W}_c^{(k)} - \bar{W}_{c+1}^{(k)} - 1 \right] p_{c+1}^{(k)} \quad (5.13)$$

and

$$g_C^{(k)} = u \left[\bar{W}_C^{(k)} - \bar{W}_1^{(k)} - 2 \right] p_1^{(k)}. \quad (5.14)$$

The transition matrix $\mathbf{G}^{(k)}$ of (5.12) has all its columns summing to unity, while its entries are non-negative and smaller than unity. As such, via the Perron–Frobenius theorem [127], we find that the maximum magnitude of all eigenvalues of $\mathbf{G}^{(k)}$ is unity, that is, all eigenvalues of any instantiation of \mathbf{G} are within (or on) the unit circle. Hence, under iterations with matrices \mathbf{G} , the system in (5.9) will converge to a steady state. All vectors

$$\mathbf{w}^{(\text{SS})} = \left[\left[\frac{W}{C} \right] \dots \left[\frac{W}{C} \right] \right]^T \quad (5.15)$$

comprise the eigenvectors of the system in (5.9) and lead to $\mathbf{G}^{(\text{SS})} = \mathbf{I}$ (i.e., they all correspond to unity eigenvalues at all time periods during the steady state). This is because all $\mathbf{w}^{(\text{SS})}$ of (5.15) lead to

$$\forall c : \begin{cases} u \left[\bar{W}_c^{(\text{SS})} - \bar{W}_{c+1}^{(\text{SS})} - 1 \right] = 0 \\ u \left[\bar{W}_C^{(\text{SS})} - \bar{W}_1^{(\text{SS})} - 2 \right] = 0 \end{cases} \\ \Rightarrow \forall c : g_c^{(\text{SS})} = 0.$$

Thus, we have

$$\forall c : \lim_{k \rightarrow \infty} W_c^{(k)} \in \left\{ \left[\frac{W}{C} \right], \left[\frac{W}{C} \right] \right\}. \quad (5.16)$$

Once C channels have balanced numbers of nodes, the DT-SCS protocol performs repeated PCO-based synchronization [123] across channels and desynchronization [122] within each channel. The former technique leads to synchronized beacon transmissions of SYNC nodes across channels, while the latter ensures fair TDMA scheduling between the nodes in a channel. The following proves the efficacy of the algorithm.

Proposition 2 *For each channel c , the proposed DT-SCS protocol converges to equidistant beacon packet transmissions at intervals of $T \left(\frac{1}{W_c} \pm b_{\text{thres}} \right)$ seconds, with $W_c \in \left\{ \lceil \frac{W}{C} \rceil, \lfloor \frac{W}{C} \rfloor \right\}$ and the SYNC beacons in all channels being broadcast concurrently.*

Proof. PCO-based synchronization is well-known to achieve convergence [123]. Hence, during the *Converging* mode, all SYNC nodes will converge to synchronous beacons across all C channels, given that their beacon packet broadcasts are only affected by other SYNC node broadcasts. PCO-based desynchronization within each channel is then equivalent to anchored desynchronization [126]. The latter is proven to converge to a steady state wherein the packet broadcasts are equidistant within the beacon period, i.e., at intervals of $T \left(\frac{1}{W_c} \pm b_{\text{thres}} \right)$ seconds. Once this is achieved and all nodes are balanced across all channels (the latter is ensured via Proposition 1), the system moves to *Converged* mode.

Channel swapping events occurring during the steady state do not affect the converged beacon packet transmissions within each channel since: (i) nodes between unbalanced channels cannot perform swaps; (ii) swapping requests and acknowledgments are done in the guard time periods; (iii) once swapping is acknowledged, nodes broadcast their first beacon packet in their new channel at the end of the guard period. In this way, they can confirm that the node they are swapping with has left the channel. Selecting the post-beacon guard period to be smaller than $b_{\text{thres}}T$ seconds ensures that the convergence is not disturbed by channel swapping.

5.3.2 Connectivity

Via channel swapping, the SYNC node in each channel c can listen to the transmissions of any node in the other remaining channels $\bar{c} \in \{1, \dots, C\}$, $\bar{c} \neq c$, except for other SYNC nodes, since they are concurrently transmitting. Hence, the degree of connectivity of a SYNC node is

$$D_{\text{SYNC}} = W - C. \quad (5.17)$$

Similarly, for all channels with $\lfloor \frac{W}{C} \rfloor$ or $\lceil \frac{W}{C} \rceil$ nodes, all DESYNC nodes can swap channels in order to listen to the transmissions of any other SYNC or DESYNC node, except for the DESYNC node that is synchronous to them. In the *Converged* mode, the

$$C_{\text{high}} = W - \left\lfloor \frac{W}{C} \right\rfloor C \quad (5.18)$$

highest channels will have

$$W_{\text{DESYNC,high}} = \left\lfloor \frac{W}{C} \right\rfloor - 1 \quad (5.19)$$

DESYNC nodes (and one SYNC node), while the

$$C_{\text{low}} = C - \left(W - \left\lfloor \frac{W}{C} \right\rfloor C \right) \quad (5.20)$$

lowest channels will have

$$W_{\text{DESYNC,low}} = \left\lfloor \frac{W}{C} \right\rfloor - 1 \quad (5.21)$$

DESYNC nodes (and one SYNC node).

Proposition 3 *The average degree of connectivity of a DESYNC node is*

$$\begin{aligned} D_{\text{DESYNC}} = & \frac{1}{W - C} [(C_{\text{high}} W_{\text{DESYNC,high}})^2 \\ & + (C_{\text{low}} W_{\text{DESYNC,low}})^2 \\ & + C_{\text{low}} C_{\text{high}} \times (W_{\text{DESYNC,high}} + W_{\text{DESYNC,low}})]. \end{aligned} \quad (5.22)$$

Proof. The average degree of connectivity of a DESYNC node is given by the total number of connections established by DESYNC nodes divided by the total number of DESYNC nodes (i.e., $W - C$). The total number of connections is found by multiplying the number of DESYNC nodes with the number of connections established by each of them. Particularly, each DESYNC node in a channel can connect to (i) all the SYNC nodes, (ii) the remaining DESYNC nodes in the same channel and (iii) the DESYNC nodes in other balanced channels (i.e., channels with the same number of nodes) that do not have the same phase. Hence, the number of connections established by DESYNC nodes in the highest and lowest channels is

$$C_{\text{high}} W_{\text{DESYNC,high}} \times (C_{\text{high}} W_{\text{DESYNC,high}} + C_{\text{low}})$$

and

$$C_{\text{low}} W_{\text{DESYNC,low}} \times (C_{\text{low}} W_{\text{DESYNC,low}} + C_{\text{high}}),$$

respectively. Summing the above expressions and dividing by $W - C$ leads to (5.22).

In the example of Fig. 5.2, we get $D_{\text{SYNC}} = 9$ and $D_{\text{DESYNC}} = 9$, while in the example of Figs. 5.5 and 5.6, we get $D_{\text{SYNC}} = 9$ and $D_{\text{DESYNC}} = 7.2$. In contrast, for the same wireless node placement as Fig. 5.6, TSCH achieves average connectivity of 3.5, as shown at the bottom of Fig. 5.1.

5.3.3 Estimation of Energy Consumption

It is generally accepted that the radio chipset draws the most power in IEEE802.15.4-based WSN deployments [121]. As an illustration, Table 5.1 presents the power requirements for different energy states of TelosB at 3 Volts [128, 129].

Table 5.1. Energy Requirements of a Crossbow TelosB Mote. The TI MSP430 MCU is the Mote Microcontroller Unit. All Values Are Reported in milli-Watt (mW). The Transmit Operation has Variable Power Levels (the reported corresponds to full power, 0 dBm)

MSP430 MCU		Chipcon CC2420 Radio chip			
Sleep	Active	Sleep	Idle	Receive	Transmit
0.0153	5.4	0.003	1.28	59.1	52.2

The energy consumption of a single node is broken down into two parts, i.e., receiving and transmitting. The energy due to receiving is analyzed for *Converging* and *Converged* mode. While the channel is in *Converging* mode, nodes listen continuously. Once converged, nodes listen to the data slots of other nodes and to swap requests. These parameters are summed to give the total amount of energy dissipated to receive by each DT-SCS node (SYNC or DESYNC), i.e.,

$$E_{\text{Rx}} = P_{\text{Rx}} \left\{ t_{\text{conv}} + n_{\text{SS}} \times \left[(W'_c - 1) \frac{T}{W_c} \right] \right\} \quad (5.23)$$

where P_{Rx} is the power of the transceiver when receiving, t_{conv} is the time required for DT-SCS to converge, n_{SS} is the number of periods the WSN operates in *Converged* (SS) mode, and W'_c is the number of nodes the node listens to.

Likewise, the energy dissipated for transmission is split into the energy used to (i) broadcast beacon packets and swap RQ/ACK and (ii) to transmit data. Assuming that each node broadcasts on average one beacon packet and one swap request or acknowledgment per period T , and that each node transmits data for every interval of $\frac{T}{W'_c}$ seconds following its beacon packet, the total amount of energy dissipated for transmission is

$$E_{\text{Tx}} = P_{\text{Tx}} \left[2 \times t_{\text{beacon}} \frac{t_{\text{conv}}}{T} + n_{\text{SS}} \times \frac{T}{W_c} \right] \quad (5.24)$$

with P_{Tx} the transmit power, t_{beacon} the time taken to transmit a beacon or swap RQ/ACK message and $\frac{t_{\text{conv}}}{T}$ the expected number of beacon transmissions until convergence.

5.4 Simulation Results

All simulations for DT-SCS were performed in Matlab [33], by extending the event-driven simulator for the DESYNC protocol by [122]. Simulations report key metrics related to events such as beacon transmission of a node, nodes joining or leaving the network, swapping requests and node balancing requirements. The results are compared against TSCH simulation results produced via the 6tisch simulator [108], which is the most accurate TSCH simulator available in the public domain.

We use data payload size of 60 bytes. Packet loss is simulated by randomly dropping packets to mimic interference conditions experienced within the 2.4 GHz unlicensed band. Simulations were repeated 100 times and average results are reported. In the vast majority of the reported results, the span of 95% confidence intervals was found to be only $\pm 15\%$ from the average values.

Concerning the configuration of the proposed DT-SCS, we used $\alpha = 0.9$, $\beta = 0.2$ for the DESYNC and SYNC parameters of (5.1) and (5.3), $T = 76$ ms, $t_{\text{swap}} = 2$ ms⁵ and $b_{\text{thres}} = 0.01$. We set $N_e = N_c = 10$ to avoid transitioning from the *Converged* to the *Converging* or *Election* mode under the simulated packet losses (see Section 5.2.3). Under the specified settings and excluding the guard time periods (15 ms), one data packet of 60 bytes can be sent within two consecutive beacon packets within the same channel.

Regarding TSCH, we use the default 6tisch settings for timeslots (101 slots) per slotframe and channels (16 channels). In our experiments, each node has, on average, two outgoing (data sending) links and one incoming (data receiving) link. In addition, the `--traffic` parameter of 6tisch is set to 0.75, which, under the established setup, corresponds to two timeslots per node link within each slotframe. Convergence is assumed for TSCH when ten consecutive slotframes are observed with less than 5% change in timeslot allocations amongst nodes.

5.4.1 Node Balancing and Connectivity

In the first simulation, we show that the proposed node balancing mechanism within DT-SCS converges to $\lfloor \frac{W}{C} \rfloor$ or $\lceil \frac{W}{C} \rceil$ nodes per channel. Figs. 5.6(a)-(b) show the initial and final node beacon-packet phases versus the channel number for $W = 14$ nodes in $C = 4$ channels. In the initial state [see Fig. 5.6(a)], a random number of nodes, each with a random phase, enter each channel. In this example, we have $W_{\{1,2,3,4\}} = [5, 3, 2, 4]$. In the converged state [see Fig. 5.6(b)], the nodes have been balanced within the channels (with the elected SYNC nodes indicated in red), where the two highest channels have four nodes and the two lowest channels have three nodes.

⁵ We opted for the reported values of T and t_{swap} such that, under the expected number of nodes per channel in steady state (i.e., 4 nodes), the duration of the data payload interval in-between the guard times becomes 15 ms, which matches the data payload interval of TSCH.

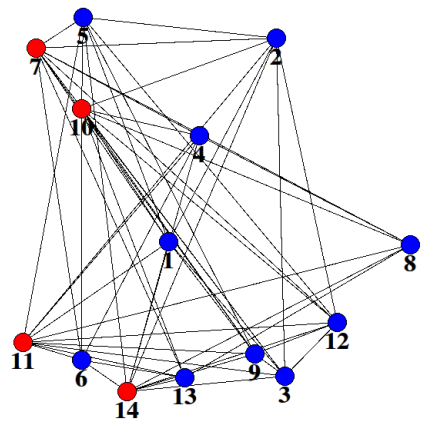
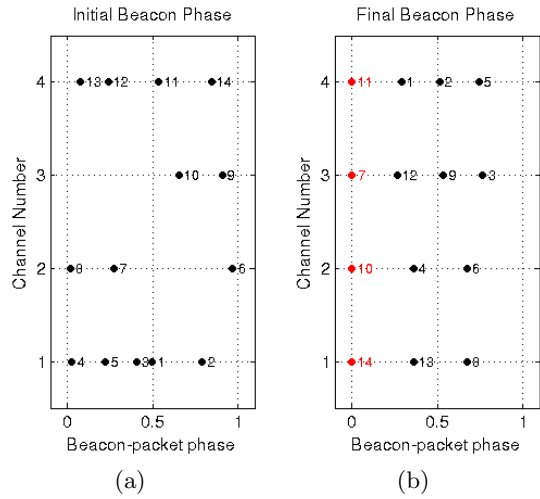


Fig. 5.6. Initial (a) and final (b) node beacon-packet phase locations versus channel number. Each node has a unique id, with SYNC nodes indicated in red. (c) Corresponding connectivity between DT-SCS nodes in the *Converged* mode, with node swapping enabled.

Fig. 5.7 illustrates the node switching between channels until balancing is achieved. Specifically, it is shown that, after 16 iterations, the channels are balanced, while, after 26 iterations, an ascending number of nodes per channel is obtained, and therefore, DT-SCS reaches the *Converged* mode. The node distribution per channel then remains stable until a node joins or leaves the network.

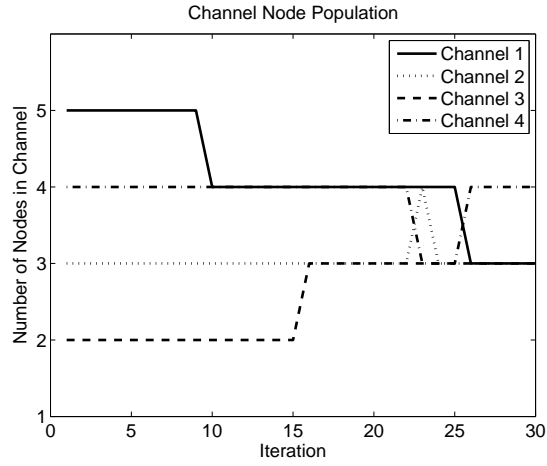


Fig. 5.7. Movement of nodes between channels as a result of the balancing mechanism.

5.4.2 Convergence Time of DT-SCS versus TSCH

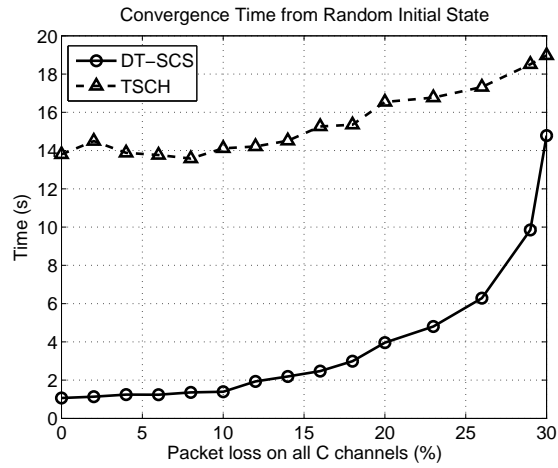


Fig. 5.8. Average time required for DT-SCS to reach *Converged* mode and for TSCH (simulated via 6tisch) to reach a stable slotframe allocation when 64 nodes join 16 channels randomly during initialization.

An important aspect of the proposed protocol is the time required to reach the *Converged* mode from a random initial state. We investigate this aspect in comparison to the time required by TSCH to achieve a stable contention-

free slot allocation via its advertising mechanism. Fig. 5.8 presents the related results under varying packet loss percentage imposed on each of the 16 channels of IEEE802.15.4. Even though these simulations do not incorporate all the aspects of propagation and interference experienced in a real testbed, the results in Fig. 5.8 demonstrate that, under disturbances in the WSN communications, DT-SCS reduces the required convergence time by 22.04–91.61% in comparison to TSCH. This is because, contrary to TSCH, the proposed DT-SCS protocol does not require nodes to advertise and acknowledge free slots, which add latency to the convergence to steady state.

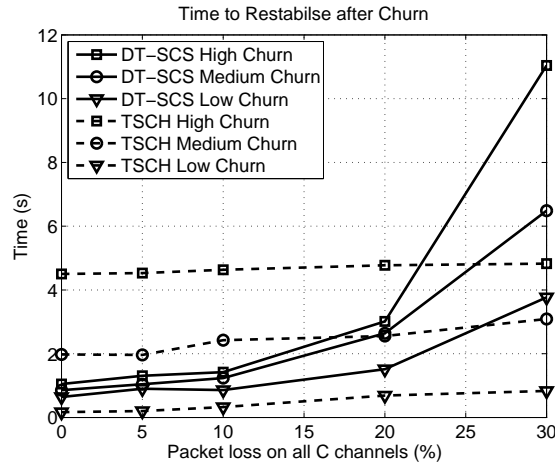


Fig. 5.9. Average time required for DT-SCS and TSCH to return to steady state under the effect of node churn.

Subsequently, we study the time required for the network to return to the steady state under the effect of churn. In this case, the WSN was initially in steady state but the arrival or departure of nodes (i.e., the effect of churn) caused the network to return to *Converging* mode. Fig. 5.9 depicts a comparison of the re-convergence speed of the proposed DT-SCS against that of TSCH for different churn conditions, namely, low, medium and high churn. These conditions correspond to 5%, 25%, and 50% of nodes arriving or leaving the WSN, respectively. The results show that, under medium and high churn and packet loss rates lower than 22–25%, DT-SCS reduces the time that the network requires to return to steady state in comparison to TSCH. This is because under medium and high churn both protocols will require extensive reconfiguration to return to steady state. Similar to the convergence from a random initial state, the proposed DT-SCS has a quicker convergence when compared to TSCH. On the contrary, TSCH offers a faster convergence when the packet loss rate is high, or when low churn is experienced. This is to be expected since, under low churn, only the TSCH nodes that have lost commu-

Table 5.2. Average Convergence Time (in seconds) Under Hidden Terminals.

	DT-SCS	TSCH
Without Hidden Terminals	1.1356	13.5845
With Hidden Terminals	1.8514	13.2957

Table 5.3. Average Convergence Time (in seconds) Under Targeted Interference.

	DT-SCS	TSCH
On a random channel ($c \neq 1$)	1.2496	14.2186
On DT-SCS ch. $c = 1$ or TSCH control ch.	1.2496	73.9126

nication links will engage in re-advertising RQ/ACK actions. Conversely, the proposed protocol will force all nodes to re-converge. Moreover, under high packet loss, few nodes receive advertising RQ/ACK, and so the schedule remains largely stable. In both cases, however, the disadvantage is that not all abandoned TSCH slots are reoccupied, thereby leading to lower bandwidth utilization.

We now investigate the convergence speed of the proposed protocol when some of the nodes in the WSN are hidden from other nodes (that is, under the effect of hidden terminals). In particular, we measure the time to achieve convergence to steady state when four randomly chosen nodes⁶ cannot communicate with a random subset of twenty other nodes in the considered WSN setup. The results in Table 5.2 show that, irrespective of the presence of hidden nodes, the convergence speed of DT-SCS is significantly higher than that of TSCH. When hidden nodes are present, the required convergence time of DT-SCS increases by 63.03%, while that of TSCH decreases slightly by 2.13%. Again, this is to be expected since TSCH nodes simply ignore RQ packets from hidden nodes. On the other hand, the *Converging* mode of DT-SCS will perform channel switching until all nodes join channels with non-hidden terminals.

Next, we study the convergence time of the proposed protocol against TSCH under the effect of targeted interference, namely, high packet losses on a given channel. In this regard, we devise the following experiment: We apply packet loss of 30% on channel \hat{c} of DT-SCS and TSCH, while all other channels $c \neq \hat{c}$ suffer from packet loss of 2%. We explore two cases, i.e., (i) when \hat{c} is a random channel ($1 \leq \hat{c} \leq 16$), or (ii) \hat{c} is the control channel of TSCH and a specific channel (e.g., $c = 1$) of DT-SCS. Contrasting the results of Tables 5.2 and 5.3 shows that the convergence time of both protocols is increased with targeted interference. However, contrary to the proposed DT-SCS, TSCH is particularly vulnerable to packet losses on the control channel, whereby the convergence time is increased by 444%. This underlines the importance of the decentralised, infrastructure-less, nature of the proposed pro-

⁶ Nodes can be SYNC or DESYNC.

ocol and highlights potential problems with centralised protocols that rely on control nodes or coordination channels. Particularly, under high control channel interference, a WSN deployment using TSCH will struggle to maintain time synchronisation across all channels, thereby suffering from a loss of performance.

5.4.3 Bandwidth Efficiency

To assess the steady state performance of the proposed DT-SCS against TSCH, we compare the total payload bits successfully received by all DT-SCS nodes per second versus the equivalent results obtained via the 6tisch simulator for TSCH. Fig. 5.10 shows that our approach achieves a substantially-higher slot and channel utilization, leading to bandwidth gains between 27.12–40.63%. This is because DT-SCS allows for all nodes to use all the available time in-between their own beacon and the next beacon (barring the guard time intervals) for contention-free transmission. On the contrary, TSCH requires advertisement and confirmation actions and imposes a rigid slot allocation. Such a rigid slotframe allocation imposes strict limitations on the available bandwidth per node operating under TSCH, restrictions that are not applied by the proposed DT-SCS protocol.

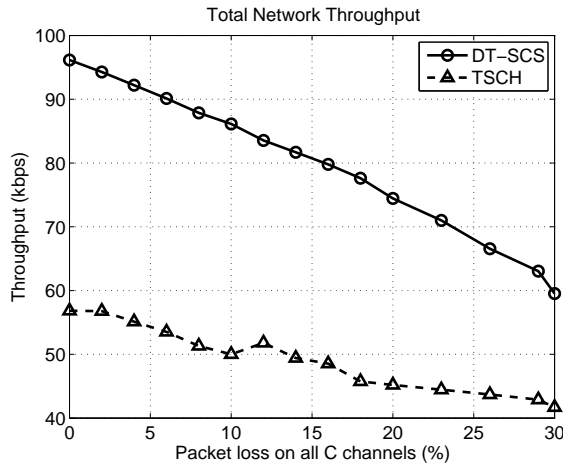


Fig. 5.10. Comparison of bandwidth utilization (total payload transmitted by all nodes per second) between the proposed DT-SCS and TSCH.

5.5 Experiments With TelosB Motes

Towards a practical evaluation testbed, we implemented both DT-SCS and TSCH as applications in the Contiki 2.6 operating system running on TelosB motes. By utilizing the NullMAC and NullRDC network stack options in Contiki, we control all node interactions at the MAC layer via our application code. We again consider a WSN deployment with $W = 64$ nodes in $c = 16$ channels, which leads to $W_c = 4$ nodes per channel in the steady state⁷. The nodes were distributed in four neighboring rooms located on the same floor, where each room contains 16 nodes plus 4 extra nodes, each of the latter configured to listen to a particular channel. These nodes were introduced for monitoring purposes and for collecting the experimental results.

An example room of our experimental setup is depicted in Fig. 5.11. We set the beacon period of the nodes to $T = 228$ ms, which is scaled by a factor of three compared to the simulations. Given the resolution⁸ of the standard Contiki timers, this value ensures that sequential node beacon broadcasts are distinguishable.



Fig. 5.11. Example of one of the rooms comprising our experimental setup. The 4 nodes at the right-most part of the image are used for monitoring purposes, while the RF signal generator, which acts to generate interference, is shown in the background.

⁷ We have experimentally verified that each node (either SYNC or DESYNC) can reach 48 other nodes via randomly-generated channel swapping requests, which agrees with (5.17) and PROPOSITION 3.

⁸ The maximum resolution of the standard Contiki callback timer on the TelosB platform is $\frac{1}{128} \simeq 7.81$ ms.

5.5.1 Energy Consumption

We measure the energy consumption of DT-SCS per node by placing a TelosB sensor running DT-SCS in series with a high-tolerance 1-Ohm resistor and by utilizing a high-frequency oscilloscope to capture the current flow through the resistor in real time. No other devices operating in the 2.4 GHz band were present in the surrounding area, hence we assume the expected packet loss to be negligible. Average results collected over 10 minutes of operation are reported. A snapshot of the oscilloscope showing the power consumption profile of a TelosB mote using DT-SCS in *Converged* mode is given in Fig. 5.12. By setting the nodes to listen for beacons only once every eight periods (instead of every period) after they reach the steady state⁹, the average power consumption without transmitting or receiving payload was measured to be 1.58 mW. This power consumption mainly stems from beacon transmission and reception. The theoretically expected value, estimated via (5.23) and (5.24) and the values of Table 5.1, was found to be 1.31 mW. This validates our implementation against our theoretical analysis. To set an illustrative comparison, the power consumption of a node operating TSCH under minimal payload (i.e., 128 bytes per four seconds) is 1.64 mW [121].

5.5.2 Results Under Interference

In the following, we investigate the convergence time of the WSN using either the proposed DT-SCS protocol or TSCH, under varying interference levels. We carried out 100 independent tests, with each room containing an interference generator for 25 tests. To generate interference, an RF signal generator (see Fig. 5.11) was used to create an unmodulated carrier in the center of each WSN channel. The carrier amplitude was adjusted to alter the signal-to-noise-ratio (SNR) at each receiver [130, 131]. The nodes were set to maximum power (+0 dBm) in order to operate under the best SNR.

Fig 5.13 shows the time required for DT-SCS and TSCH to converge under varying interfering signal power levels. The results obtained with our Contiki implementation corroborate that the proposed DT-SCS reduces the convergence time by an order of magnitude in comparison to TSCH. Moreover, the difference in convergence time between the proposed protocol and TSCH is increasing with the interference level. This underlines the key advantages of the DT-SCS protocol with respect to TSCH, namely, the fact that it (i) is fully decentralized and (ii) does not depend on an advertisement and acknowledgement scheme.

Next, we investigate the convergence time of the proposed DT-SCS protocol and TSCH under the effect of targeted interference on a given channel.

⁹ We have confirmed experimentally that such "sparse listening" mode during SS does not affect the stability of DT-SCS under standard residential or office test conditions.

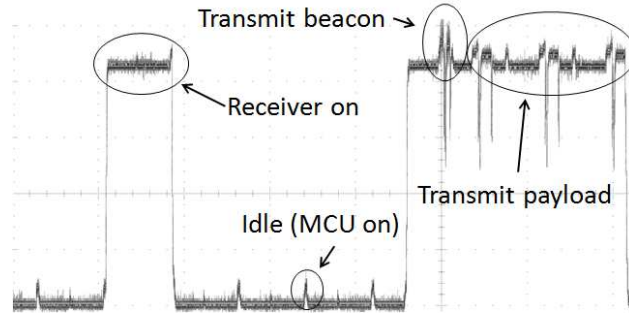


Fig. 5.12. Oscilloscope snapshot depicting the power consumption of a TelosB mote under the proposed DT-SCS. When no payload is transmitted, energy is consumed by the processor (MCU) and the radio chipset that transmits and listens for beacons.

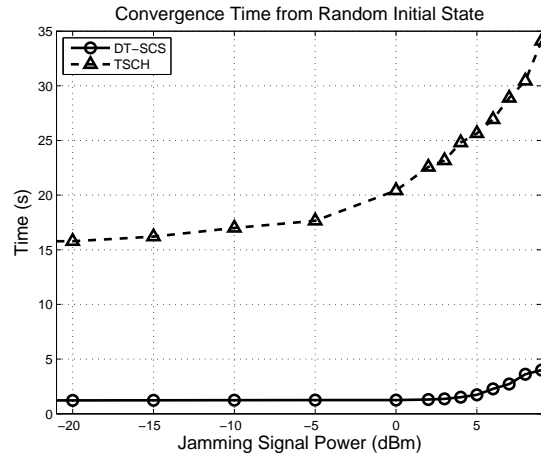


Fig. 5.13. Average time required for DT-SCS to reach *Converged* mode and for TSCH to reach a stable slotframe allocation under varying interfering signal power levels. Both DT-SCS and TSCH were implemented as an application in the Contiki 2.6 operating system running on TelosB motes.

Concerning the former, given that there is no coordination channel, we explore how the interference on channel $c + 1$ effects the convergence in channel c . A moderate level of interference (that is, 5 dBm) in channel $c + 1$ causes fluctuations in the SYNC node beacon of channel c , which in turn causes the convergence time to increase. Particularly, in this case, the convergence time of the nodes in the observed channel c is increased from 1.223 to 1.518 seconds. When the same level of interference is also applied on the observed channel, the convergence time is further increased to 2.738 seconds. Regarding TSCH, we observed that interference in the advertisement channel led to unstable behavior and, for the cases where convergence was achieved, more than 30

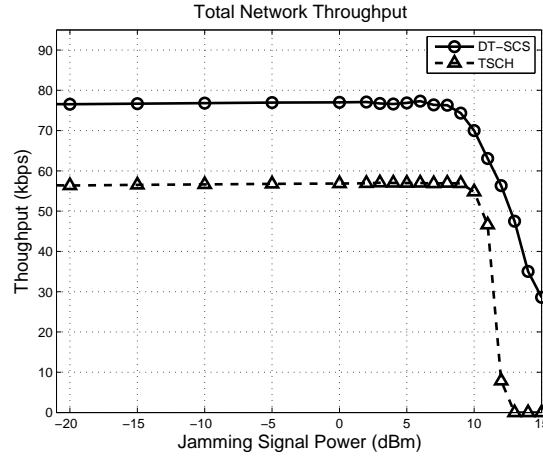


Fig. 5.14. Total network bandwidth (total payload bits transmitted by all nodes per second) between the proposed DT-SCS and TSCH under varying signal power levels. Both DT-SCS and TSCH were implemented as an application in the Contiki 2.6 operating system running on TelosB motes.

seconds were required. This demonstrates experimentally the detriment of depending on a coordination channel for advertisement.

5.5.3 Bandwidth Results

Subsequently, we measure the total network bandwidth (that is, total payload bits per second)—under the same interference conditions—achieved with the proposed DT-SCS and TSCH. The results, depicted in Fig. 5.14, show that DT-SCS systematically achieves more than 40% increase in the total network throughput, irrespective of the interference level. Both protocols experience a significant loss of throughput under high interference levels (that is, above 10 dBm), which is, however, substantially more severe for TSCH. In effect, when interference is above 12 dBm, the bandwidth obtained with TSCH drops to zero. This is due to the inability of TSCH to recover lost slots through advertising, since almost all advertisement and acknowledgment packets are lost due to interference. On the contrary, even at high interference levels, DT-SCS can recuperate bandwidth utilization due to the elasticity of SYNC and DESYNC mechanisms, discussed in Section 5.3.

The experimental results on TelosB motes confirm that the proposed DT-SCS protocol offers higher bandwidth utilization and more robustness to interference compared with the TSCH protocol that is included in the IEEE802.15.4e-2012 standard. Moreover, the proposed protocol allows for quicker convergence times with respect to TSCH. Therefore, DT-SCS is shown to be an excellent candidate for real-life WSN deployments that call for (i) high

bandwidth (e.g., visual sensor networks), (ii) decentralized, infrastructure-less, network reconfiguration and fast convergence, and (iii) robustness to high levels of interference.

5.6 Conclusion

We propose a novel protocol for WSNs that performs decentralized time-synchronized channel swapping (DT-SCS) and circumvents certain convergence and network utilization problems of existing designs, such as the state-of-the-art TSCH protocol of IEEE802.15.4e-2012. The unique aspect of our approach is the use of pulsed coupled oscillators that *concurrently* perform synchronization and desynchronization in *multiple channels*. This allows for rapid convergence to the steady state in a completely decentralized manner, that is, without requiring a node or channel coordinator, or time synchronization via a global clock. DT-SCS spontaneously adapts to node churn and varying packet losses, while offering high degree of connectivity through channel swapping. Experimentation via simulations and a real Contiki-based implementation on TelosB motes shows that, in comparison to TSCH, the proposed DT-SCS leads to a significant reduction of the convergence time and substantially higher network throughput utilization. These traits render the proposed DT-SCS an excellent candidate for WSN deployments that collect and communicate large quantities of information in a decentralized manner.

This work led to the writing and publication of the following work:

G. Smart, N. Deligiannis, Y. Andreopoulos, R. Surace, V. Loscrí, G. Fortino, “**Decentralized Time-Synchronized Channel Swapping for Wireless Sensor Networks**”, poster publication in *11th European Conference on Wireless Sensor Networks (EWSN 2014)*, February 2014.

Conclusion

This PhD thesis has focused on the Self-Organizing Systems. The work done has shown as such devices need to be programmed in a new way through methodologies that allow the learning process and the emerge of behaviors that fit with the network issues.

Methodologies useful for this purpose as we have shown, are evolving neural networks inspired by biology and the mobility of the network components. Through this new concept the network becomes a Self-Organized System where the components act in a cooperative way through simple locale rules learned from the environment and interactions with other nodes and the global behavior that emerges responds adequately to the dynamics of the surrounding environment that can also be particularly hostile.

In this PhD thesis we showed a background on Self-Organizing Systems. We investigated on the impact of the Propagation Environment on Controlled Mobility Algorithms; proposed and tested distributed heuristics to Film Sport Events with Flying Robots. Proposed and tested Bio-Inspired approaches to solve multi-objective problem. In this context, we used the algorithm proposed for the video surveillance through ultra-low power sensors. Finally, we proposed and tested a new communications protocol for WSN called Decentralized Time-Synchronized Channel Swapping.

References

1. Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraulazand, Eric Bonabeau, “**Self-Organization in Biological Systems**”, in *Princeton University Press*, 2001.
2. A. Kansal, M. Rahimi, D. Estrin, W.J. Kaiser, G.J. Pottie, M.B. Srivastava, “**Controlled Mobility for Sustainable Wireless Sensor Networks**”, in *Proceedings of IEEE SECON*, Page(s):1-6, 2004.
3. Collier, Travis , Taylor, “**Self-organization in sensor networks**”, in *Journal Parallel Distrib. Comput.*, vol. 64, issue 7, pages 866-873, July 2004.
4. Falko Dressler, “**Self-Organization in Ad-Hoc Networks: Overview, Classification**”, in *University of Erlangen, Dept. of Computer Science 7, Technical Report 02/06*.
5. W. Elmenreich, R. D’Souza, C. Bettstetter, H. De Meer, “**A Survey of Models, Design Methods for Self-Organizing Networked Systems**”, in *Proceedings of the 4th International Workshop on Self-Organizing Systems*, 2009.
6. M. Bhatt, R. Chokshi, S. Desai, S. Panichpapiboon, N. Wisitpongphan, 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.
7. I. F. Akyildiz, I. H. Kasimoglu, “**Wireless Sensor, Actor Networks: Research Challenges**”, in *Ad Hoc Networks Journal (Elsevier)*, Vol. 2, No. 4, pp. 351-367, October 2004.
8. A. Sanfeliu, N. Hagita, A. Saffiotti, “**Network robot systems**”, in *Robotics, Autonomous Systems Journal*, vol. 56, number 10, pages 793-797, 2008.
9. M. Quaritsch, E. Stojanovski, C. Bettstetter, G. Friedrich, H. Hellwagner, B. Rinner, M. Hofbauer, M. Shah, “**Collaborative Microdrones: Applications, Research Challenges**”, in *Proceedings of the 2nd International Conference on Autonomic Computing, Communication Systems*, pages 38:1-38:7, 2008.
10. E. Yanmaz, C. Bettstetter, “**Area Coverage with Unmanned Vehicles: A Belief-Based Approach**”, in *Proc. IEEE Vehicular Technology Conf. (VTC)*, pp. 1-5, 2010.
11. M. Grossglauser, D. Tse, “**Mobility Increases the Capacity of Ad Hoc Wireless Networks**”, in *IEEE/ACM Trans. on Networking*, vol 10, no 4, August 2002.

12. Ryo Sugihara and Rajesh K. Gupta, “**Speed control and scheduling of data mules in sensor networks**”, in *ACM Transactions on Sensor Networks (TOSN) TOSN Homepage archive, Volume 7 Issue 1, August 2010 ACM New York, NY, USA*.
13. 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*.
14. Mohammad Rafiqul Haider and Syed Kamrul Islam, “**Sensors and Low Power Signal Processing**”, in *Springer-Verlag New York Inc. January 2010*.
15. F. Bullo, R. Carli, and P. Frasca, “**Gossip Coverage Control for Robotic Networks: Dynamical Systems on the Space of Partitions**”, in *SIAM Journal on Control and Optimization, July 2011*.
16. Z. Butler and D. Rus, “**Controlling Mobile Sensors for Monitoring Events with Coverage Constraint**”, in *Proceedings of IEEE ICRA, Volume:2, Page(s):1568-1573, 2004*.
17. 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*.
18. 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*.
19. 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*.
20. 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*.
21. 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*.
22. 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*.
23. 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*.
24. 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*.
25. Carmelo Costanzo, Enrico Natalizio and Valeria Loscri, “**Distributed Virtual-Movement Scheme for Improving Energy Efficiency in Wireless Sensor Networks**”, in *MSWiM 2009: 297-304, Canary-Islands (Spain), October, 2009*.
26. E. Natalizio and V. Loscri, “**Controlled Mobility in Mobile Sensor Networks: Advantages, Issues and Challenges**”, in Springer Telecommunication Systems, in *Special Issue on Recent Advance in Mobile Sensor Networks, DOI: 10.1007/s11235-011-9561*.

27. Di WU, Lichun Bao and Renfa Li, “**A holistic approach to wireless sensor network routing in underground tunnel environments**”, in *Computer Communications, Volume 33, Issue 13, August, 2010*.
28. Sun Kun, Wang Ping and Li Yingze, “**Path Loss Models for Suburban Scenario at 2.3GHz, 2.6GHz and 3.5GHz**”, in *Antennas, Propagation and EM Theory, 2008. ISAPE 2008, 8th International Symposium on DOI: 10.1109/IS-APE.2008.4735242 Publication Year: 2008 , Page(s): 438 - 441*.
29. E. L. Lehmann and G. Casella, “**Theory of Point Estimation**”, in *New York: Springer, 1998*.
30. Wendi Beth Heinzelman, “**Application-Specific Protocol Architectures for Wirelless Network**”, submitted to the *Department of Electrical Engineering and Computer Science for the degree of Doctor of Philosophy at the Massachusetts Institue of Technology*.
31. T. Rappaport, “**Wireless Communications: Principles and Practice**”, in *Pearson Education, 2010*.
32. Paul A. Tipler, Todd Ruskell and Gene Mosca, “**Physics for Scientists and Engineers**”, in *W. H. Freeman, 2008*.
33. Matlab <http://www.mathworks.it/products/matlab/>
34. F. Daniyal, A. Cavallaro, “**Multi-camera scheduling for video production**”, in *Euorpean Conference on Visual Media Production (CVMP), Nov. 2011*.
35. F. Z. Qureshi, D. Terzopoulos, “**Surveillance in virtual reality: system design and multi-camera control**”, in *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition, pages 1-8, June 2007*.
36. X. Zhou, R.T. Collins, T. Kanade, P. Metes, “**A master-slave system to acquire biometric imagery of humans at distance**”, in *ACM International Workshop on Video Surveillance, 113-120, ACM Press, 2003*.
37. P. Halvorsen, S. Sægrov, A. Mortensen, D.K.C. Kristensen, A. Eichhorn, M. Stenhaug, S. Dahl, H.K. Stensland, V.R. Gaddam, C. Griwodz, D. Johansen, “**Bagadus: an integrated system for arena sports analytics: a soccer case study**”, in *Proc. of the International Conference on Multimedia Systems (MMSys), 48-59, February/March 2013*.
38. F. Guerriero, R. Surace, V. Loscrí, E. Natalizio, “**A Multi-objective Approach for Unmanned Aerial Vehicle Routing Problem with Soft Time Windows Constraints**”, in *Applied Mathematical Modelling, Volume 38, Issue 3, 1 February 2014, Pages 839-852*.
39. P. Kilby, P. Prosser, P. Shaw, “**Dynamic VRPs: A Study of Scenarios**”, in *University of Strathclyde Technical Report, APES-06-1998, 1-11, 1998*.
40. M. Pavone, “**Dynamic Vehicle Routing for Robotic Networks**”, in *Ph.D. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2010*.
41. F. Bullo, E. Frazzoli, M. Pavone, K. Savla, S. L. Smith, “**Dynamic Vehicle Routing for Robotic Systems**”, in *IEEE Proceeding, 99(9): 1482-1504, 2011*.
42. S. Pudlewski, T. Melodia, “**C-DMRC: Compressive Distortion-Minimizing Rate Control for Wireless Multimedia Sensor Networks**”, in *Proc. of IEEE Intl. Conf. on Sensor and Ad-hoc Communications and Networks (SECON), June 2010*.
43. T. Melodia, I. F. Akyildiz, “**Cross-layer QoS-Aware Communication for Ultra Wide Band Wireless Multimedia Sensor Networks**”, in *IEEE Journal of Selected Areas in Communications, vol. 28, p. 653-663, June 2010*.

44. J. Rullán-Lara, S. Salazar, R. Lozano, “**UAV real-time location using a Wireless Sensor Network**”, in *Proceedings of 8th Workshop on Positioning Navigation and Communication (WPNC)*, p. 18-23, April 2011.
45. E. Natalizio, R. Surace, V. Loscrí, F. Guerriero, T. Melodia, “**Two Families of Algorithms to Film Sport Events with Flying Robots**”, in *10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2013)*, October 2013.
46. M. Quaritsch, K. Kruggl, D.W.Strucl, S. Bhattacharya, M. Shah, B. Rinner, “**Networked UAVs as aerial sensor network for disaster management applications**”, in *Elektrotechnik und Informationstechnik*, n. 127, v. 1, p. 56-63, January 2010.
47. Y. Xu, B. Lei, E. A. Hendriks, “**Camera network coverage improving by particle swarm optimization**”, in *J. Image Video Process.*, January 2011.
48. D.K. Ahner, A.H. Buss, J. Ruck, “**Assignment Scheduling Capability for UAV - A Discrete Event Simulation with Optimization in the Loop Approach to Solving a Scheduling Problem**”, in *Proc. of the Winter Simulation Conference*, December 2006.
49. M. Ehrgott, X. Gandibleux, “**A survey and annotated bibliography of multiobjective combinatorial optimization**”, in *OR Spektrum*, v. 22, p. 425-460, 2000.
50. V. Chankong, Y.Y. Haimes, “**Multiobjective Decision Making Theory and Methodology**”, in *Elsevier Science, New York*, 1983.
51. H.N. Psaraftis, “**Dynamic vehicle routing problems**”, in *Elsevier Science*, pp. 223-248, 1998.
52. **LINGO** <http://www.lindo.com/>.
53. J.R. Schott, “**Fault tolerant design using single and multicriteria genetic optimization**”, in *Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts*, 1995.
54. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “**A fast and elitist multiobjective genetic algorithm: NSGA-II**”, in *IEEE Transactions on Evolutionary Computation*, n. 2, v. 6, p. 182-197, 2002.
55. G. Aloï, L. Bedogni, M. Di Felice, V. Loscrí, A. Molinaro, E. Natalizio, P. Pace, G. Ruggeri, A. Trotta, N. R. Zema, “**STEM-NET: An Evolutionary Network Architecture for Smart and Sustainable Cities**”, in *Transactions on Emerging Telecommunication Technologies, Special Issues on Smart Cities*, Vol. 25, Issue 1, pages 21, 40, January 2014.
56. M. Di Felice, L. Bedogni, A. Trotta, L. Bononi, F. Panziera, G. Ruggeri, G. Aloï, V. Loscrí, P. Pace, “**Smartphones Like Stem Cells: Cooperation and Evolution for Emergency Communication in Post-Disaster Scenarios**”, in *Proceedings of IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom 2013)*, July 3-5, 2013, Batumi, Georgia.
57. C. Blümm, C. Heller, R. Weigel, “**SDR OFDM Waveform Design for a UGV/UAV Communication Scenario**”, in *Journal of Signal Processing Systems*, Vol. 69, Issue 1, pp 11-21, October 2012.
58. V. Pejovic, E. M. Belding, “**WhiteRate: A Context-Aware Approach to Wireless Rate Adaptation**”, in *IEEE Transactions on Mobile Computing*, Volume 13, Issue 4, April 2014, pp. 921-934.
59. B. B. Godbole, D. S. Aldar, “**Performance Improvement by Changing Modulation Methods for Software Defined Radios**”, in *International Journal of Advanced Computer Science and Applications*, 1(6), Dec. 2010.

60. P. Pace, G. Aloï, “**Disaster monitoring and mitigation using aerospace technologies and integrated telecommunication networks**”, in *IEEE Aerospace and Electronic Systems Magazine*, 23(4), April 2008.
61. V. Loscrí, P. Pace, R. Surace, “**Multi-objective evolving neural network supporting SDR modulations management**”, in *Proceedings of 24th IEEE Personal Indoor Mobile and Radio Communications (PIMRC)*, 8-11 September, London-UK, 2013.
62. I. Oka, M.P.C. Fossorier, “**A General Orthogonal Modulation Model for Software Radio**”, in *IEEE Transactions Communications*, vol. 54, No.1, Jan. 2006, pp. 7-12.
63. E. Marpanaji et al., “**Experimental Study of DQPSK Modulation on SDR Platform**”, in *ITB Journal of Information and Communication Technology*, vol. 1, No.2, 2007, pp. 84-98.
64. **GNU Radio** <http://gnuradio.org>. - Accessed on March. 2, 2013.
65. **Ettus Research LLC** <http://www.ettus.com> - Accessed on March. 2, 2013.
66. T. R. Múck, A. A. Fróhlich, “**HYRA: A Software-defined Radio Architecture for Wireless Embedded Systems**”, in *10th International Conference on Networks (ICN)*, January, 23-28, 2011.
67. C. Costanzo, V. Loscrí, E. Natalizio, T. Razafindralambo, “**Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network**”, in *Computer Communications*, 35(9), 2012.
68. G. Molina, E. Alba, E. Talbi, “**Optimal Sensor Network Layout Using Multi-Objective Metaheuristics**”, in *Journal of Universal Computer Science*, n 15, pp 2549-2565, vol. 14, 2008.
69. R. Surace, V. Loscrí, E. Natalizio, “**On the impact of the propagation environment on controlled mobility algorithms**”, in *International Workshop on Mobility and Communication for Cooperation and Coordination (MC3) at IEEE ICNC 2012*, pp. 62-66, 2012.
70. S. Butman, B. Levitt, I. Bar-David, R. Lyon, M. Klass, “**Design Criteria for Noncoherent Gaussian Channels with MFSK Signaling and Coding**”, in *IEEE Transactions on Communications*, vol. 24, No. 10, 1976, pp. 1078-1088.
71. L. Hanzo, S. Ng, T. Keller, W. Webb, “**Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems**”, in *John Wiley*, 2004.
72. F.M. Costa, H. Ochiai, “**A comparison of modulations for energy optimization in wireless sensor network links**”, in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp 1-5, December 2010.
73. S. Cui, A.J. Goldsmith, A. Bahai, “**Energy-constrained modulation optimization**”, in *IEEE Transactions on Wireless Communications*, 4(5):2349-2360, September 2005.
74. **Roving Networks wireless for less WiFly GX 802.11G Super Module: RN-121, RN-122, RN-123, RN-125** <http://www.rovingnetworks.com>.
75. A. Goldsmith, “**Wireless Communications**”, in *Cambridge University Press*, August 2005.
76. M. Michal, “**A survey of Multi-Objective deployment in Wireless Sensor Networks**”, in *Journal of Telecommunications and Information Technology*, vol. 2010, issue 3, pp. 36-41, 2010.

77. S. R. Goyal, “**A Survey on deployment Methods in Wireless Sensor Networks**”, in *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, issue 7, July 2013.
78. L. P. Damuut, D. Gu, “**Survey of Deterministic Vs Non-Deterministic Node Placement Schemes in WSNs**”, in *sixth International Conference on Sensor Technologies and Applications, SENSORCOMM 2012*.
79. P. Stuckmann, R. Zimmermann, “**Toward ubiquitous and unlimited-capacity communication networks: European research in Framework Programme 7**”, in *IEEE Communications Magazine*, vol. 45, Issue 5, May, 2007.
80. B. B. Godbole, D. S. Aldar, “**Performance Improvement by Changing Modulation Methods for Software Defined Radios**”, in *International Journal of Advanced Computer Science and Applications*, 1(6), Dec. 2010.
81. R. R. Bhambare, R. D. Raut, “**A Survey on Digital Modulation Techniques for Software Defined Radio Applications**”, in *International Journal of Computer Networks and Wireless Communications (IJCNWC)*, ISSN: 2250-3501, vol. 3, No 3, June 2013.
82. **FREVO** <http://www.frevotool.tk>.
83. M. Magno, P. Zappi, D. Brunelli and L. Benini, “**Solar-powered Video Sensor Node for Energy Efficient Multimodal Surveillance**”, in *11th EURO-MICRO Conference on Digital System Design, 2008*.
84. T. Yan, T. He, and J. A. Stankovic, “**Differentiated surveillance for sensor networks**”, in *SenSys, 2003*, pp. 51-62.
85. Oh, P. Chen, M. Manzo, and S. Sastry, “**Instrumenting wireless sensor networks for real-time surveillance**”, in *proc. of the International Conference on Robotics and Automation, May, 2006*.
86. D. Mendez, A. J. Perez, M. A. Labrador, J. J. Marron, “**P-Sense: a participatory sensing system for air pollution monitoring and control**”, in *Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 344-347, March 2011.
87. **PIR** <http://www.ladyada.net/learn/sensors/pir.html>.
88. M. Magno, D. Boyle, D. Brunelli, E. Popovici, L. Benini, “**Ensuring Survivability of Resource-Intensive Sensor Networks Through Ultra-Low Power Overlays**”, in *IEEE Transactions on Industrial Informatics*, vol.10, no.2, pp.946,956, May 2014.
89. M. Magno, F. Tombari, D. Brunelli, L. Di Stefano, L. Benini, “**Multimodal Video Analysis on Self-Powered Resource-Limited Wireless Smart Camera**”, in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol.3, no.2, pp.223,235, June 2013.
90. V. Jelicic, M. Magno, D. Brunelli, V. Bilas, L. Benini, “**Benefits of Wake-up Radio in Energy-Efficient Multimodal Surveillance Wireless Sensor Network**”, in *IEEE Sensors Journal*, no.99, pp.1,1, 2014.
91. A. Iera, G. Ruggeri, D. Tripodi, “**Providing Throughput Guarantees in 802.11e WLAN Through a Dynamic Priority Assignment Mechanism**”, in *Special Issue on Advances in Wireless LANs and PANs, Kluwer Wireless Pers. Commun. J.*, 2005.
92. M. Magno, F. Tombari, D. Brunelli, L. Di Stefano, L. Benini, “**Multimodal Abandoned/Removed Object Detection for Low Power Video Surveillance Systems**”, in *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS '09*, pp.188-93, Sept. 2009.

93. A. Kerhet, F. Leonardi, A. Boni, P. Lombardo, M. Magno, L. Benini, “**Distributed video surveillance using hardware-friendly sparse large margin classifiers**”, in *IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007*, pp.87-92, Sept. 2007.
94. I.F. Akyildiz, T. Melodia, K.R. Chowdhury, “**A survey on wireless multimedia sensor networks**”, in *Computer networks*, n. 4, p. 921-960, vol. 51, 2007.
95. Y.C. Tseng, Y.C. Wang, K.Y. Cheng, Y.Y. Hsieh, “**iMouse: an integrated mobile surveillance and wireless sensor system**”, in *IEEE Computer*, n. 6, p. 60-66, vol. 40, 2007.
96. N. Deligiannis, F. Verbist, J. Slowack, R. Van de Walle, P. Schelkens, A. Munteanu, “**Progressively refined Wyner-Ziv video coding for visual sensors**”, in *ACM Transactions on Sensor Networks*, n. 2, p. 21, vol. 10, 2014.
97. J.J. Acevedo, B. Arrue, I. Maza, A. Ollero, “**Distributed Approach for Coverage and Patrolling Missions with a Team of Heterogeneous Aerial Robots Under Communication Constraints**”, in *International Journal on Advanced Robotic Systems*, n. 28, vol.10, 2013.
98. L. Merino, A. Gilbert, J. Capitán, R. Bowden, J. Illingworth, A. Ollero, “**Data fusion in ubiquitous networked robot systems for urban services**”, in *Annales des Telecommunications*, n. 7-8, p. 355-375, vol. 67, 2012.
99. N. Deligiannis, F. Verbist, J. Barbarien, J. Slowack, R. Van de Walle, P. Schelkens, A. Munteanu, “**Distributed coding of endoscopic video**”, in *IEEE International Conference on Image Processing (ICIP’11)*, p. 1813-1816, 2011.
100. S. Pollin, M. Ergen, M. Timmers, A. Dejonghe, L. Van der Perre, F. Catthoor, I. Moerman, A. Bahai, “**Distributed cognitive coexistence of 802.15. 4 with 802.11**”, in *International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, p. 1-5, 2006.
101. T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, K. Pister, “**OpenWSN: a standards-based low-power wireless development environment**”, in *Transactions on Emerging Telecommunications Technologies*, n. 5, p. 480-493, vol. 23, 2012.
102. G. Wu, S. Talwar, K. Johnsson, N. Himayat, K.D. Johnson, “**M2M: From mobile to embedded internet**”, in *IEEE Communications Magazine*, n. 4, p. 36-43, vol. 49, 2011.
103. Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, M. Guizani, “**Home M2M networks: architectures, standards, and QoS improvement**”, in *IEEE Communications Magazine*, n. 4, p. 44-52, vol. 49, 2011.
104. L. Atzori, A. Iera, Antonio, G. Morabito, “**The internet of things: A survey**”, in *Computer networks*, n. 15, p. 2787-2805, vol. 54, 2010.
105. G. Mulligan, “**The 6LoWPAN architecture**”, in *Proc. 4th Workshop on Embedded Networked Sensors*, p. 78-82, 2007.
106. A. Tinka, T. Watteyne, K. Pister, “**A decentralized scheduling algorithm for time synchronized channel hopping**”, in *Springer Ad Hoc Networks*, p. 201-216, 2010.
107. P. Zand, S. Chatterjea, K. Das, P. Havinga, “**Wireless industrial monitoring and control networks: The journey so far and the road ahead**”, in *Journal of Sensor and Actuator Networks*, n. 2, p. 123-152, vol. 1, 2012.
108. Q. Wang, X. Vilajosana, T. Watteyne, “**6TSCH Operation Sublayer (6top)**”, 2013.

109. H.K. Le, D. Henriksson, T. Abdelzaher, “**A practical multi-channel media access control protocol for wireless sensor networks**”, in *Proc. of the 7th international conference on Information processing in sensor networks*, p. 70-81, 2008.
110. S. M. George, W. Zhou, H. Chenji, M. Won, Y.O. Lee, A. Pazarloglou, R. Stoleru, P. Barooah, “**DistressNet: a wireless ad hoc and sensor network architecture for situation management in disaster response**”, in *IEEE Communications Magazine*, n. 3, p. 128-136, vol. 48, 2010.
111. D. Buranapanichkit, Y. Andreopoulos, “**Distributed Time-Frequency Division Multiple Access Protocol for Wireless Sensor Networks**”, in *IEEE Wirel. Comm. Lett.*, n. 5, p. 440-443, vol. 1, Oct. 2012.
112. J. Hwang, T. Kim, J. So, H. Lim, “**A receiver-centric multi-channel MAC protocol for wireless networks**”, in *Computer Communications*, 2012.
113. L. Tang, Y. Sun, O. Gurewitz, D.B. Johnson, “**EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks**”, in *Proc. 12th ACM Int. Symp. on Mob. Ad Hoc Netw. and Comput.*, p. 23:1-23:11, 2011.
114. Y. Kim, H. Shin, H. Cha, “**Y-MAC: An energy-efficient multi-channel mac protocol for dense wireless sensor networks**”, in *Proc. of the 7th international conference on Information processing in sensor networks*, p. 53-63, 2008.
115. M. Miller, N.H. Vaidya, “**A MAC protocol to reduce sensor network energy consumption using a wakeup radio**”, in *IEEE Transactions on Mobile Computing*, n. 3, p. 228-242, vol. 4, 2005.
116. M. Ali, T. Suleman, Z.A. Uzmi, “**MMAC: A mobility-adaptive, collision-free mac protocol for wireless sensor networks**”, in *IEEE International Performance, Computing, and Communications Conference (IPCCC'05)*, p. 401-407, 2005.
117. T. Luo, M. Motani, V. Srinivasan, “**CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks**”, in *International Conference on Broadband Communications, Networks and Systems (BROADNETS'06)*, p. 1-10, 2006.
118. J. Borms, K. Steenhaut, B. Lemmens, “**Low-overhead dynamic multi-channel mac for wireless sensor networks**”, in *Wireless Sensor Networks*, p. 81-96, 2010.
119. K. Pister, L. Doherty, “**TSMP: Time synchronized mesh protocol**”, in *IASTED Distr. Sensor Netw.*, 391-398, 2008.
120. J. Song, S. Han, A.K. Mok, D. Chen, M. Lucas, M. Nixon, W. Pratt, “**WirelessHART: Applying wireless technology in real-time industrial process control**”, in *IEEE Real-Time and Embed. Tech. and Appl. Symp.*, (RTAS'08), p. 377-386, 2008.
121. X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, K. Pister, “**A Realistic Energy Consumption Model for TSCH Networks**”, in *IEEE Sensor Journal*, 2013.
122. J. Degesys, I. Rose, A. Patel, R. Nagpal, “**DESYNC: self-organizing desynchronization and TDMA on wireless sensor networks**”, in *Proc. 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*, p. 11-20, 2007.

123. R. Pagliari, Y.W. Hong, A. Scaglione, “**Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling**”, in *IEEE J. on Select. Areas in Commun.*, n. 4, p. 564-575, vol. 28, May 2010.
124. J. Degeysys, R. Nagpal, “**Towards desynchronization of multi-hop topologies**”, in *Proc. IEEE Internat. Conf. on Self-Adaptive and Self-Organizing Syst. (SASO'08)*, p. 129-138, 2008.
125. A. Patel, J. Degeysys, R. Nagpal, “**Desynchronization: The theory of self-organizing algorithms for round-robin scheduling**”, in *Proc. IEEE Internat. Conf. on Self-Adaptive and Self-Organizing Syst. (SASO'07)*, 2007.
126. C.M. Lien, S.H. Chang, C.S. Chang, D.S. Lee, “**Anchored desynchronization**”, in *Proc. IEEE INFOCOM'12*, p. 2966-2970, 2012.
127. V.G. Kulkarni, “**Modeling and analysis of stochastic systems**”, in *CRC Press*, vol. 36, 1995.
128. **Chipcon AS**, www.ti.com/product/cc2420.
129. **Crossbow Technology Inc.**, www.willow.co.uk/Telos.
130. C.A. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, A. Willig, “**Controllable radio interference for experimental and testing purposes in wireless sensor networks**”, in *IEEE 34th Conference on Local Computer Networks (LCN'09)*, p. 865-872, 2009.
131. C.A. Boano, T. Voigt, C. Noda, k. Romer, K M Zuniga, “**Jamlab: Augmenting sensor network testbeds with realistic and controlled interference generation**”, in *10th International Conference on Information Processing in Sensor Networks (IPSN'11)*, p. 175-186, 2011.

References of the Candidate

- P1. R. Surace, V. Loscrí, E. Natalizio, “**On the Impact of the Propagation Environment on Controlled Mobility Algorithms**”, in *International Workshop on Mobility and Communication for Cooperation and Coordination (MC3) at IEEE ICNC 2012*.
- P2. V. Loscrí, P. Pace, R. Surace, “**Multi-Objective Evolving Neural Network supporting SDR Modulations Management**”, in *24th IEEE International Symposium on Personal, Indoor, Mobile and Radio Communications (PIMRC 2013), September 2013*.
- P3. E. Natalizio, R. Surace, V. Loscrí, F. Guerriero, T. Melodia, “**Two Families of Algorithms to Film Sport Events with Flying Robots**”, in *10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2013), October 2013*.
- P4. F. Guerriero, R. Surace, V. Loscrí, E. Natalizio, “**A Multi-objective Approach for Unmanned Aerial Vehicle Routing Problem with Soft Time Windows Constraints**”, in *Applied Mathematical Modelling, Volume 38, Issue 3, 1 February 2014, Pages 839-852*.
- P5. G. Smart, N. Deligiannis, Y. Andreopoulos, R. Surace, V. Loscrí, G. Fortino, “**Decentralized Time-Synchronized Channel Swapping for Wireless Sensor Networks**”, poster publication in *11th European Conference on Wireless Sensor Networks (EWSN 2014), February 2014*.
- P6. V. Loscrí, M. Magno, R. Surace, “**Video Surveillance Applications based on Ultra-Low Power Sensor**”, in *1th International Workshop on Autonomous Monitoring and Networking (WAMN 2014), August 2014*.