MOHAMED KHIDER BISKRA UNIVERSITY

MASTER THESIS

Genetic Algorithm-based Optimal Wireless Sensor Network Deployment

Author:
WAFA BOUAZIZ

Supervisor:
MRs.Yousra BENAISSA

A thesis submitted in fulfillment of the requirements for the degree of MASTER

in the

THE FACULTY OF EXACT SCIENCES NATURAL AND LIFE SCIENCES
Department of computer science

July 13, 2022

Acknowledgements

First of all, I would like to thank Allah and say Al-Hamdulillah for the good life, good health, supreme protection, strength, help, and guidance throughout my studies.

I would also like to thank my project supervisor, **Dr. Yousra Benaissa** , for guiding and helping me during this battle and for her helpful instructions and suggestions to be a better person and accomplish my work, as well as for the trust she has always shown me, and for inviting me to the "LINFI" lab to complete my memo there.

I also thank **Mrs. Mohammedi Amira** for accepting to compensate my supervisor on the day of the discussion.
All the thanks to the teachers in computer science at the University of Biskra, Specially the amazing teacher **Dr, Bitam Salim**, And for those who helped me in one way or another to accomplish this memo.

I would like to express my gratitude to my friends and colleagues who provided me with their material, moral and intellectual support, and valuable comments that enabled me to overcome difficulties and get advanced in my studies and research.

I owe a special thanks to my family, who are my small world for their ultimate support and encouragement, for those who accompanied me on this journey, and also for standing with me against all the obstacles and hard times I faced.

All the thanks to my parents for their love, encouragement, and support, for their prayers and duaa, A special thanks to my mother, the lady of my life, the number one supporter and straightener when weak, I appreciate all her sacrifices, To my father, the man of my life, my supporter in my studies since childhood, The author of the saying, "You only study, and what remains is on me."

My special thanks go to my brother and sisters, with whom we shared home, life, secrets, and everything, All thanks to the great of the family, to the far but near in our hearts always, to my sister **Rogaia**, To **Abdelbasset**, my only brother who, no matter how pampered and my many demands, he is always here for me.

I owe thanks to my beloved, my soul mate, my second mother, and my partner in crime, and to the one who shaped me to face this life, **Fatima**

(Asmaa). To my roommate, to my likeness till the point where people confuse us, **Chaima**. And to the one, I took care of in childhood, to the distinguished **Aya**. And to the last cluster **Nada**.

Big thanks to them all, for their valuable support, I owe them forever for giving me chances and experiences that made me who I am today, this journey wouldn't be, if not for them.
Without forgetting to thank my nieces and nephews for their love, I forced them to be absolutely silent during my study and preparation for this memorandum.

Without them, I would not be here.

Thank you all.

I gift this humble work to those whom nobody can make up for the sacrifices they made for my education and my well-being :

This letter is mainly to my dear parents, ***Souâd*** *and* ***Abderrahmane****, Who raised me with love, supported me, and encouraged me during the years of study and all my life, may Allah protect them.*

To my sisters and my only brother, who were always there for me, who handled me and stood by my side.

To the first of grandsons of the ***Al Bouaziz*** *family, my beautiful who always fills me with love* ***Sirine****, to who makes me smile in my hard times my doll* ***Sidra*** *, And to the troublemaker* ***Islem****, And for those who will come later.*

For every person who was by my side on this trip, To any person who gave me the courage to move forward.

For all who loved me and whom I love.

BOUAZIZ Wafa,

Abstract

Since the need for environmental monitoring and event detection has increased, many wireless sensor network (WSN) applications have achieved significant advances in sensor node technology. Sensor nodes are able to communicate and sense environmental conditions, however, deploying these nodes can be expensive and dangerous in some cases, which leads to random deployment as the only possible solution. Hence, the main issue of WSN is to find the near-optimal deployment in all environmental circumstances, since random deployment may lead to uncovered or unconnected areas, duplicated or missed data, and to waste more energy. Therefore, a Genetic Algorithm (GA) based methodology is implemented for self-organized WSN. In order to implement the process of WSN deployment and ensure low energy consumption, an optimization method is required. Based on the experimental results, GA helps in identifying the best position for each sensor node in order to meet network connectivity and coverage.

Keywords: Wireless Sensor Network, Sensors Deployment, Network Connectivity and Coverage, Multi-objective Optimization, Genetic Algorithm.

Résumé

Depuis que le besoin de surveillance environnementale et de détection d'événements a augmenté, de nombreuses applications de réseaux de capteurs sans fil (RCSF) ont réalisé des progrès importants dans la technologie des nœuds de capteurs. Les nœuds de capteurs sont capables de communiquer et de détecter les conditions environnementales, mais le déploiement de ces nœuds peut être coûteux et dangereux dans certains cas, ce qui conduit à un déploiement aléatoire comme seule solution possible. Par conséquent, le principal problème de RCSF est de trouver le déploiement presque optimal dans toutes les circonstances environnementales, puisque le déploiement aléatoire peut conduire à des zones non couvertes ou non connectées, des données dupliquées ou manquées, et à gaspiller plus d'énergie. Par conséquent, une méthodologie basée sur un algorithme génétique (AG) est mise en œuvre pour les RCSF auto-organisés. Afin de mettre en œuvre le processus de déploiement de RCSF et d'assurer une faible consommation d'énergie, une méthode d'optimisation est nécessaire. Sur la base des résultats expérimentaux, GA aide à identifier la meilleure position pour chaque nœud de capteur afin de répondre à la connectivité et la couverture du réseau.

Mots-clés: Réseau de Capteurs Sans Fil, Déploiement de Capteurs, Connectivité et Couverture Réseau, Optimisation Multi-Objectifs, Algorithme Génétique.

Contents

List of Figures

List of Tables

List of Abbreviations

WSN	**W**ireless **S**ensor **N**etwork
GA	**G**netic **A**lgorithm
BS	**B**ase **S**tation
PER	**P**acket **E**rror **R**ate
ADC	**A**nalog **D**igital **C**onverters
UDP-Like	**U**ser **D**atagram **P**rotocol **L**ike
RWS	**R**oulette **W**heel **S**election **L**ike

List of Symbols

P_r	Received Power
P_s	Sensitivity Power
P_t	Transmission Power Radiated by the transmitter
T	Temperature
R_c	Communication Range
A	Area
N	Number of Nodes
C_c	Connectivity
C_v	Coverage
v	speed

General Introduction

Sensor technologies and wireless communication have undergone a wonderful evolution in recent years and have become one of the most controversial that have attracted the attention of researchers. Where some of them have dubbed the first decade of the twenty-first century the "decade of sensors". This is due to the large number of domains using sensors. In fact, the sensors are triggering a revolution similar to that experienced by microcomputers in the 1980s. Since the need for environmental monitoring and event detection increased, huge applications have gotten advancements in sensor technology [7].

This evolution has allowed the emergence of small electronic devices called sensor nodes, with very low cost and limited energy resources. These sensor nodes are capable of collecting and processing information in an autonomous and flexible way [8]. They can be also deployed and interconnected on a large scale, providing a new type of network called Wireless Sensor Networks (WSN). WSN has been used in an extensive diversity of utilization and has appeared as an auspicious study field. Where it has been utilized in diverse industrial and civilian application areas such as industrial process monitoring and control, machine health monitoring, smart home-field, and environmental monitoring. These sensor nodes are typically equipped with communication and data processing skills to assemble data and track information toward a base station (BS) [8].

Therefore, the primary objective of the target protocols is to maximize the number of targets that can be covered in the application field. The most important coverage issue is the deployment strategy used to distribute sensor nodes in the application field. Where, sensor nodes can be deployed either manually or randomly by dropping them from an aircraft. However, Random deployment is generally preferred in large-scale WSNs, not only because it is easy and less expensive, but also because it may be the only choice in remote and hostile environments [9]. Moreover, the power source of these sensor nodes includes a limited energy budget battery which outcomes in determining nodes' lifetime, as well as in determining WSN lifetime. Additionally, it may be unlikely or difficult to recharge these batteries as sensor nodes may be organized in an aggressive or impracticable environment.

Due to the external causes or intended by the system designers, the WSNs may change dynamically, which may affect network connectivity, localization, coverage, as well as QoS. Nevertheless, the traditional approaches of WSNs are explicitly programmed, so as a result, the network does not work properly in the dynamic environment. Hence, managing such a large number of nodes requires a scalable and efficient algorithm.

One of the most important algorithms that were suggested in the field of WSNs is the Genetic algorithm, which is one of the most powerful heuristics for solving optimization problems that is based on natural selection. This process drives biological evolution by the construction of a fitness function.

In this thesis, we have proposed to generate the near-optimal deployment in WSN by a Genetic Algorithm and modify it in order to obtain an effective solution.

Thesis organization

This thesis is organized into four chapters and is presented in the following form.

- Chapter 1: This chapter introduces the wireless sensor network and its applications, where it describes the main concepts of sensor networks and exposes the basic notions to well understand the studied problem, such as the hardware and software architecture of sensor node. it begins by presenting the Wireless Sensor Networks, their characteristics, and the most important applications. This chapter goes over various types of wireless network deployment modeling.

- Chapter 2: This chapter describes the different genetic operators such as genetic mutation and crossover. As well as it presents a comparative study of related works, methods, and used metrics in this field.

- Chapter 3: This chapter designs the global architecture of genetic algorithms that are considered as one of the most efficient optimization techniques. While it performs some modifications to make the near-optimal deployment dynamic in order to meet network connectivity and coverage under temperature variations.

- Chapter 4: this chapter describes the various technical aspects related to the proposed system's implementation and deployment, starting with the evaluation of the proposed model in the context of a large-scale deployment, and to demonstrate the efficiency and applicability in realistic scenarios. After that it presents and discusses the theoretical part of the proposed project and the obtained results.

- Finally, we conclude this thesis by a global conclusion that recalls the techniques used in our system, it also presents our future prospects for this project.

Chapter 1

General information on wireless sensor networks

1.1 Introduction

The performance of a wireless network is primarily determined by the deployment process. The deployment process determines the optimal number and placement of the network's various components. The various performance metrics are the number of nodes, cost, coverage, and connectivity, which are primarily determined by the positioning of the nodes as well as the network topology used [6].

In this chapter, we present the concepts and aspects related to our research area and expose the basic notions that will lead us to well understand the studied problem. We will begin by presenting the Wireless Sensor Networks, their WSN characteristics, and the most important applications. We will also go over the various types of wireless network deployment modeling. Finally, we define some concepts related to sensor node deployment type as well as deployment criteria.

1.2 Wireless Sensor Networks (WSN)

In recent years, "wireless sensor networks" has come to prominent topic be one of the most controversial that gained the researchers' attention. In this section, we present a general overview of WSN, including its architecture, features, and application domain.

1.2.1 Definition of a WSN

A wireless sensor network (WSN) is one of the most promising technologies for some real-time applications because of its size, cost-effectiveness, and easily deployable nature. Consist of a large number of battery-powered sensor nodes outfitted with sensing, processing, storage, and wireless radio communication capabilities [7]. These nodes are typically deployed in specific areas to monitor an area and collect data about the environment, capture, process, and transmit critical data in real-time with high resolution, which is then transmitted to the base station (sink node)for post-data analysis or act accordingly.

The deployments differ depending on the application, but in general, nodes are unattended and must operate for long periods without external intervention or assistance [8]. As a result, nodes must consume the least amount of energy while ensuring that the sensor network continues to function properly. In its most basic form, the network can only function properly if both coverage and connectivity are preserved [7] [8].

Coverage ensures that the entire deployment area is monitored, ensuring that no event is missed, and connectivity ensures that the network is not fragmented, allowing data transmitted to reach any node in the network (see figure 1.1).

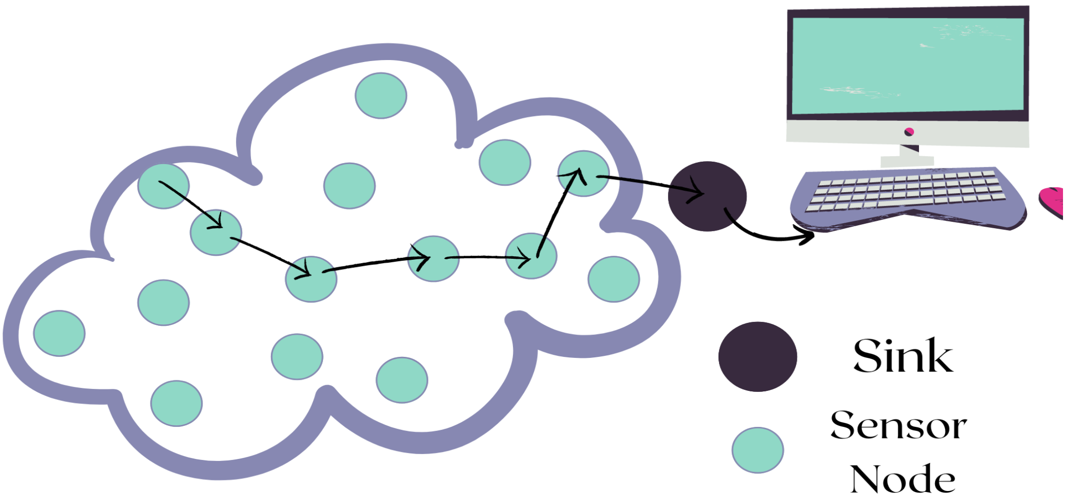

FIGURE 1.1: Architecture of Wireless Sensor Network

1.3 Sensor Node

1.3.1 Definition

A sensor node, also known as a mote (primarily in North America), is a node in a sensor network about a device designed to measure environmental data such as temperature, capture an image, and so on, transform it into usable information, and then transmit it in an analog or digital manner to a processing unit. Each sensor has three main functions: collecting sensory information, processing, and communicating with other network nodes to send the information to one or more collection points (sinks) [9].

Sensor nodes are so economical that thousands of these can be installed in the preferred locations, monitoring, and information collection device that converts one physical quantity to another of a different nature (very often electrical). This quantity, which is a representation of the quantity taken, can be used for measurement or control [10].

1.3.2 Types of sensor nodes

Sensor nodes are classified into two types: static nodes and mobile nodes.

- **Static sensor nodes** Static sensor nodes are stationary nodes that remain in the same location where they were initially deployed. They should make the most of their fixed positions and provide the best possible coverage of the target area [11].
- **Mobile sensor nodes** For better coverage, mobile sensor nodes can move and adjust their position. Algorithms can be used to determine the best node configuration to provide the greatest coverage of the area of interest [11].

1.3.3 Sensor node anatomy

To get closer to how a WSN is built, an insight into a sensor node is to come first. Specifically, a sensor node is made up of four units (a power unit, a sensing unit, a processing unit, and a communication unit) and two systems (Mobility system, Environmental location system) [1]. Each one corresponds to a specific data acquisition, processing, or transmission task. It also has an energy source. shown in figure 1.2.

There are several models on the market, among the most famous, the "mote" MICAx and TelosBe from Crossbow.

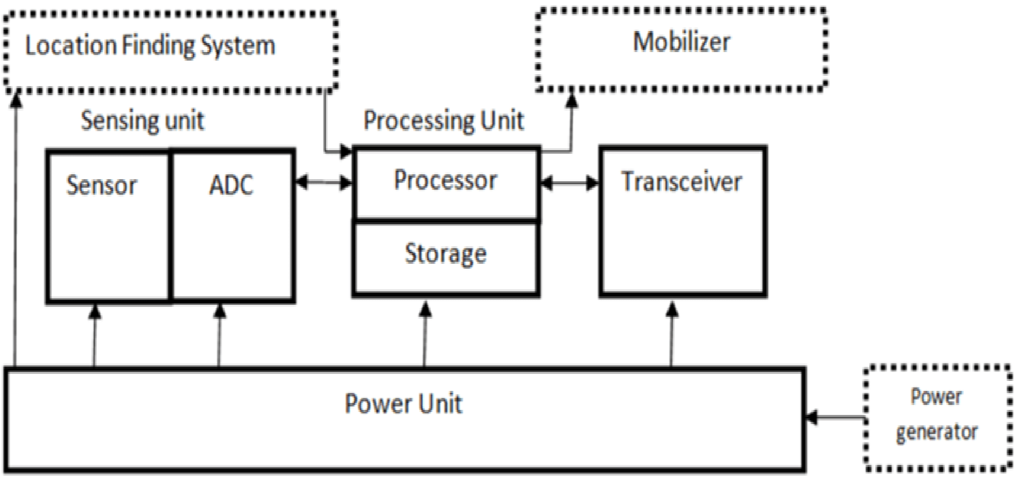

FIGURE 1.2: General architecture of wireless sensor node [1] [2]

- **Sensing units(Acquisition)**

  Are usually composed of two subunits: sensors (recognizing the analyte) and ADCs (Analog Digital Converters). Based on the observed phenomenon, the sensors obtain digital measurements of environmental parameters and transform them into analog signals, which the ADCs then convert into digital signals understandable by the processing unit and feed into the processing unit [12] [13].

- **The communication unit (transceiver unit)**

  Is responsible for connecting the node to the network. This unit, which consists of a transmitter/receiver that allows the network's sensor nodes to communicate with one another via radio links, is in charge of all data transmission and reception via a wireless communication medium. It is one of the most energy-hungry components on board and can be of optical type (are robust with regard to the electric interferences) or radio-frequency type (include modulation, demodulation, filtering, and multiplexing circuits) [12] [13].

The acquisition unit is the physical core that allows the measurement to be taken, whereas the communication unit is responsible for data transmission to other electronic devices.

- **Processing unit**

  This is a small computer that already has a processor, some memory, and general-purpose input/output ports. It is made up of two interfaces (an interface with the acquisition unit and another with the communication unit). It performs all computation on the sensor nodes and controls all procedures that allow a sensor node to collaborate with other nodes in order to carry out the tasks of data acquisition and storage [13] [14].

- **Power unit**

  Every sensor node requires energy to function. It can be a rechargeable battery, a solar rechargeable battery, or an energy harvester. It is impossible to charge or recharge a battery in sensitive environments. As a result, the energy supplied by sensor batteries is one of the most valuable resources because it has a direct impact on the lifetime of the sensors and thus on a sensor network [13] [14].

- **Location finding system**

  Detection tasks and routing techniques often need to know the geographical location of a node. Thus, the node is equipped with a geographic location system. [15].

- **Mobilizer system**

  Mobility is sometimes required to allow a node to move around in order to complete its tasks. Mobility assistance necessitates a large amount of energy, which must be provided efficiently. To control the node's movements, the mobility system can also work in close collaboration with the sensing unit and the processor [15] [16].

Each sensor node has a sensing range (Rs) and a communication range (Rc). The areas defined by these two ranges are depicted in Figure 1.3. The sensor node's communication range (Rc) is the range over which it can communicate with other nodes. The sensing range (Rs) refers to the range within which the sensor node can detect the event.

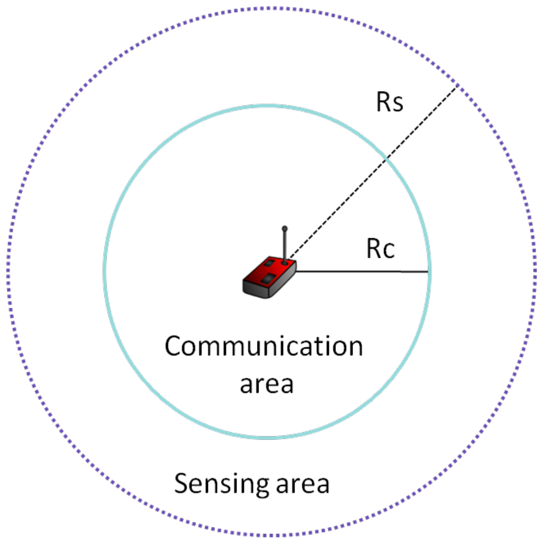

FIGURE 1.3: Sensor node Communication (Rc) and Sensing range (Rs)

All of these subunits may need to fit into a matchbox-sized module that is smaller than a cubic centimeter and light enough to float in the air. In addition to size, sensor nodes must meet a number of other stringent requirements [14] [16].

- Consume extremely low power.
- Operate at high volumetric densities.

- Have low production costs.
- Can be easily replaced, and the failure of one sensor does not prevent the operation of the others.
- Are self-sufficient and operate unattended.
- Adapt to their surroundings.

1.4 WSN characteristics

A wireless sensor network is characterised by (see [17]):

- A significant number of nodes
- Wireless access: Interference is unavoidable (disrupted radio links in a hospital).
- Limited resources (Computing (4MHz), energy (AAA batteries), and memory (512-1MB).
- Power management (battery-powered, no batteries will be changed, different standby modes (Idle Mode, Power Down Mode)).
- Deployment mode (Deployment in the wild, placed in a specific location, randomly scattered).
- Types of communication [18]:
  - Unicast(between two nodes on the network)
  - Broadcast(Sink transmits information to all nodes on the network)
  - Local Gossip(nodes located in a specific region collaborate together to have a better estimation of the observed event and to avoid sending the same message to the "Sink" node which contributes to consuming less energy)
  - Convergecast (between a group of nodes and a specific node, which can be the "Sink", saves energy at the receiving node)
  - Multicast (communication between a node and a group of nodes)

1.5 WSN Applications

WSNs are gaining immense attention from researchers due to their enormous applications in different areas shown in figure 1.4, we cite:

1.5.1 Environmental Monitoring Based on WSN Technologies

A terrestrial WSN-based environmental monitoring system's general architecture consists of hundreds or thousands of low-cost sensor nodes located in strategic points of the monitored site and whose positions are assumed to be known, as well as one or more monitoring centers responsible for collecting and processing all data acquired by the sensor nodes. They are classified based on their application fields (air, land, or sea) [19].

WSN for Air Monitoring

WSN technologies have gotten a lot of attention in the field of air environmental monitoring. With the evolving reality of smart cities, monitoring the quality of the air in all its aspects in densely populated urban areas is becoming a major concern. Several multilayer architectures are presented that enable distributed monitoring of air environmental parameters with a small number of deployed sensors.

WSN for Land Monitoring

WSN technologies have found several application contexts in the field of soil monitoring. The ever-increasing urbanization of modern society is causing a significant increase in the total amount of waste produced per year, necessitating the installation of adequate monitoring systems to oversee the storage, treatment, and recycling processes. The extreme environment is a common challenge. High temperatures, hazardous gases, and the possible presence of chemical acids in the vicinity of sensors can severely impair the WSN's correct sensing.

WSN for Marine and Water Monitoring

Over the last few decades, marine environments have been severely threatened by anthropogenic activities such as tourism, urbanization, and industry. WSNs have significantly improved real-time analyses and monitoring of marine and coastal areas. The increased sensing resolution promotes a faster response to unexpected critical events such as flooding or water contamination. A marine monitoring system is made up of a number of nodes that are placed near the coast or at strategic points on the sea surface [19].

1.5.2 Military applications

Military research was an initial driver for the development of sensor networks. Because of their low cost, lack of cabling, self-configuration, and fault tolerance, this new area of embedded computing has been very successful. The sensors are deployed autonomously to assist the military in their missions by monitoring the enemy's activities or analyzing the battlefield prior to sending troops there to detect potential threats (detection of biological, nuclear, and chemical threats) [20] [10].

FIGURE 1.4: Applications of WSN

1.5.3 Medical Applications

By embedding sensor nodes in living organisms, physicians can monitor vital functions in real-time. Also, multi-sensor capsules or micro-cameras can be swallowed without surgery and transmit images of the inside of a human body in real-time of a part of the body for approximately 24 hours. Other WSN-based biomedical applications include: -Early cancer detection, -Monitoring vital signs and activity levels in elderly or disabled people's homes, - Hospital monitoring etc [21].

1.5.4 Agriculture applications

WSNs can be deployed in agricultural fields and incorporated into the soil to measure variables such as humidity, temperature, and field condition (for example, to determine the driest areas). We can also imagine equipping herds of cattle with sensors to track their location at all times. This intelligent agriculture, also known as "precision agriculture" has for main objectives: to optimize crop inputs in order to increase yield, and the fight against food insecurity, particularly in developing countries [22] [13].

1.5.5 Commercial applications

WSNs can be used in the commercial field to assist merchants in improving the storage and delivery of goods by knowing the position, status, and direction of a given good at all times. As a result, a customer who is waiting for a specific item will be able to receive a delivery notification in real-time. As a result, businesses can provide better service while lowering costs [13].

1.5.6 Detection of natural disasters

By dispersing sensor nodes in the wild, an autonomous network can be created. Sensors can thus detect events like forest fires, storms, and floods. This allows for much faster and more efficient emergency services intervention. The use of sensor networks in the surveillance of civil structures (buildings, bridges, roads, and aircraft) can significantly reduce the financial costs associated with their monitoring. As currently, monitored with expensive and time-consuming technologies such as X-rays and ultrasound. [22] [10].

1.6 Architecture of wireless sensor networks

1.6.1 Communication architecture of an WSN

A wireless sensor network is a subset of an Ad Hoc network. It is typically composed of a large number of small sensors distributed across a geographical area known as a sensor field in order to monitor a physical phenomenon and collect data autonomously. The size of the coverage area determines the number of sensors deployed. The sensor nodes use wireless communication to route captured data to their neighbors until it reaches the base station or sink node, which serves as a data collection point. It can send the collected data to the user via a communication network, such as the internet or satellite. In turn, the user can use the base station as a gateway in order to send requests to the network [18] [23].

1.6.2 Protocol architecture

The ISO organization proposes a functional division of the entire process of communicating a network with a seven-layer structure (the physical layer, the link layer, the network layer, the transport layer, the session layer, the presentation layer, and the application layer). The OSI model appears to be overly complex for WSN. In fact, only a few applications require a seven-layer division. The network nodes, including the base station, operate on a five-layer model that corresponds to the OSI layers: the application layer, the transport layer, the network layer, the data link layer, and the physical layer [18].

The purpose of a WSN is not only communication, but it is also subject to severe energy constraints. As a result, other units must be added to manage energy consumption, node mobility, and task scheduling. The WSN also employs three plans for this purpose: energy management, mobility, and task plans. Figure 1.5 depicts this.

These management plans are critical to the operation of sensor nodes because they allow sensor nodes to collaborate in an efficient manner, allowing sensor nodes to coordinate tasks and reduce energy consumption. As a result, sensor nodes must collaborate with one another in order to route data and share resources while making efficient use of available energy. Therefore, the network's lifetime can be extended in this manner [23] [10].

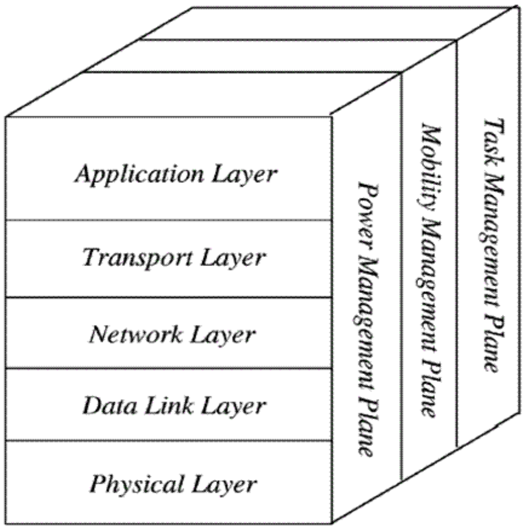

FIGURE 1.5: Layered Protocol architecture

The following sections describe the role of the various layers for the above model, as well as those of the various management layers.

Physical layer

Specifies the hardware characteristics, modulation techniques, carrier detection, digital, analog, and optical signal conversion, and allows data to be modulated and routed in physical media while selecting the appropriate frequencies [18] [13].

Data link layer

Is divided into two sub-layers: the (LLC) sub-layer provides the majority of the error management mechanisms, while the (MAC) sub-layer manages access to the physical medium by avoiding collisions. This layer is also in charge of data multiplexing and transmission error detection [13] [18].It enables control of the logical link. It also specifies how data is sent between two sensor nodes separated by one hop [23].

Network layer

The main role of the network layer is to manage the addressing and routing of the data provided by the transport layer and establish reliable routes between sensor nodes and the base station, to choose the best path in terms of energy, transmission delay, and throughput [23] [13] [18].

Transport layer

This layer is in charge of transporting data, splitting it into packets, controlling the flow, maintaining packet order, and ensuring correct data routing and transmission quality. Transmission reliability is not guaranteed in WSNs. As a result, a transport protocol similar to UDP called UDP-Like is used. However, because the TCP transport protocol is universal when communicating with an external network, the WSNs must have a TCP-splitting interface to ensure compatibility between these two communicating networks [18].

Couche application

The highest layer. It ensures interfacing with the applications. This layer is the level closest to the users. The application layer can also manage the aggregation of data before it is transferred to the transport layer [13]. Among the application protocols: SMP (Sensor Management Protocol) and TADAP (Task Assignment and Data Advertisement Protocol) [18].

The role of each of the different management layers is described in the following paragraphs [13] [18] [23]:

Task management plane

Allows for the assignment of tasks to sensor nodes, some sensor nodes can perform capture tasks while others go into power-saving mode to conserve energy. As a result, is not required that all sensor nodes belonging to the same zone are obliged to capture tasks at the same time.

Mobility management plane

Detects and records sensor node movement/mobility and keeps track of node location during the routing phase. Sensor nodes can learn who their neighbors are by using these locations, and sensor nodes can better coordinate and manage the use of their energy to perform various tasks. Due to the destruction of some nodes, node self-organization is sometimes required. The mobility management layer must be able to change the position of the nodes in this case.

Power management plane

The energy management layer governs how the sensor node uses energy. For example, after receiving a message, the receiving node's radio module can be turned off to conserve energy. When a node's energy level is low, it can broadcast a message to her neighbor sensors to avoid routing tasks and save the remaining energy for acquisition functionality.

1.7 Deployment of the wireless sensor network

Sensor node deployment is a critical concept in the setup of a wireless sensor network. This section defines sensor node deployment, as well as its various types and criteria.

1.7.1 Definition

Deployment of wireless sensor network nodes is the phase of selecting the location of sensor nodes in the environment to be monitored. This is a critical phase that significantly affects the performance of the entire network: monitoring quality, connectivity, energy consumption, and network lifetime [24].

The deployment can be either deterministic, random, or dynamic [25].

Random deployment

In this deployment, sensor nodes are scattered randomly, either by helicopter, aircraft, grenade launcher, drone, or cluster bomb. These deployment methods lead to a random distribution of sensor nodes, creating an infrastructure in an ad hoc manner. This deployment does not ensure the connectivity and coverage of the network. Indeed, the difficulty of a deterministic deployment arises in two situations [25]:

- when the number of sensors is very large.
- and/or when the deployment environment is not completely accessible.

Often, random deployment is the only possible option. For example, in battlefield reconnaissance and surveillance, disaster recovery, and forest fire detection missions, deterministic deployment of nodes is very risky and not feasible at all, see figure 1.6.

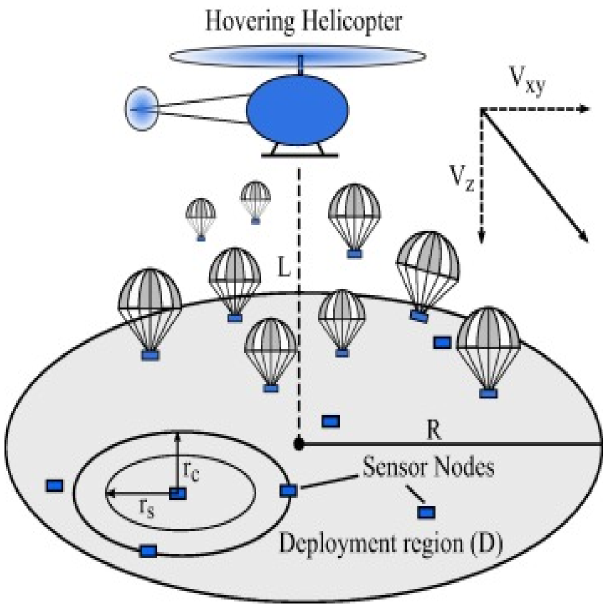

FIGURE 1.6: Sensor nodes dropping by hovering helicopter [3]

Deterministic deployment

In deterministic deployment, sensors are placed one by one in pre-determined locations, either by a human or a robot, to ensure specific performance. This deployment method is used when the cost of sensors is very high or when their operation is heavily influenced by their position. The coverage of the capture field can be ensured in this type of deployment by careful planning of node density; thus, the network topology can be established at the time of WSN installation [25].

Dynamic deployment

A static deployment strategy decides on sensor locations when the network is configured and does not account for dynamic changes during the network's operational phase. The usage of an application may change over time as new sensors join the network and the available network resources change, or if many sensors near a functional Sink fail due to battery depletion, redundant sensors in other parts of the monitored region can be identified and redeployed to replace the sensors, extending network lifetime and ensuring service continuity. These examples demonstrate that static deployment does not address many WSN applications. [25].

1.7.2 Deployment criteria

The ideal in sensor node deployment is for the deployed network to meet all design constraints, resulting in a long network lifetime. Most deployment methods in the literature focus on specific criteria such as connectivity, coverage, number of nodes, Cost, and energy consumption.

Connectivity

After the nodes have been placed in the area of interest, they must form a network in order to transfer the information captured by the source nodes to the base station [13].

A WSN node is said to be connected if and only if it can communicate directly (one-hop connectivity) or indirectly (multi-hop connectivity). Connectivity is thus primarily determined by the existence of routes, the WSN is connected if there is at least one route connecting each node in the network to the base station. It is influenced by topology changes, which are typically caused by node failures, mobility, and so on. These topology changes have an impact on the network. Because topology changes result in communication link loss, node isolation, network partitioning, and coverage loss, it is critical to carefully study and account for connectivity properties when designing and deploying such networks [26]. The distance between two nodes must be less than or equal to R_c in order to occur communication.

Coverage

Coverage is one of the most important factors to consider when deploying a WSN because it is a key performance metric. It indicates how well a given area is monitored (controlled), that is, how well each point in the monitoring area is observed and tracked by all sensor nodes. As a result, the concept of coverage in WSNs can be viewed as a measure of QoS [27].

Coverage can be defined and classified based on the density level of the node. If only parts of the area are covered by nodes, the coverage is sparse; if nodes cover the entire area, the coverage is dense; and if multiple nodes cover the same detected location, the coverage is called [28]. The coverage of an area is itself expressible in the form of several models the binary model and the probabilistic model [29].

- **Binary Coverage Model (BM)**

  The simplest detection model, the "binary disc" model. This model assumes that a node can detect only phenomena that are within its detection range. Each node's detection range is bounded by a circular disk of range "R_s" known as the detection range. This model assumes that an event is detected with a probability p by a node N of "1" (100 percent detection) if it occurs at a distance less than or equal to the detection radius R_s, and a probability of "0" if it occurs at a distance greater than or equal to the detection radius R_s (no detection at all), d(N,p) is the Euclidean distance between position p and node N as:

$$P = \begin{cases} 1 & if \quad d(N,p) < R_s \\ 0 & Otherwise \end{cases} \tag{1.1}$$

- **Probabilistic Coverage Model (PDM)**

  A probabilistic detection model proposed by (Zou and Chakrabarty) in [30]is a more realistic extension of the "binary disc" model. The coverage probability of a point P by a node N is given as follows:

$$P(n_i, n_j) = \begin{cases} 1 & if \quad d(N,p) \leq R_s \\ exp^{-\alpha d(N,p)} & if \quad R_c \geq d(N,p) > R_s \\ 0 & if \quad d(N,p) \geq R_c \end{cases} \tag{1.2}$$

  Where R_s and R_c are determined by the type of sensor node and environmental factors, is a parameter related to the node's hardware components (represents the obstacle-related energy distortion factor). All points are said to be covered by a given node if they are within a distance of (R_s), which is similar to the binary model's detection range, and all those within the interval $[R_c, R_s]$ have a coverage (<1) that decreases exponentially with increasing Euclidean distance. All points are revealed beyond a certain distance (R_c).

Energy consumption

The consumption of power is a significant constraint in the design and implementation of WSN. In most cases, replacing the battery is impossible. This means that the lifespan of a sensor is highly dependent on the battery's lifespan. Each node in a (multi-hop) sensor network collects data and sends values. The failure of some nodes necessitates a change in network topology and packet rerouting. All of these operations require a significant amount of energy. As a result, current research is primarily concerned with ways to reduce this consumption [31].

1.8 Conclusion

In this chapter, we have tried to give a general view of WSNs, where we have laid down the basic building blocks and federated some necessary concepts. The WSN is one of the great interest which represents a technological revolution in measuring instruments, these sensor networks open up incalculable development possibilities in all application areas. This allows us to believe that sensor networks will soon become an integral part of our lives, surely satisfying the needs of our largest projects.

In addition, due to the external causes or intended by the system designers, the WSNs may change dynamically. Therefore it may affect network routing strategies, localization, coverage, and QoS, but the traditional approaches for the WSNs are explicitly programmed, and as a result, the network does not work properly for the dynamic environment. Therefore managing such a large number of nodes requires scalable and efficient algorithms. In the next chapter, we will focus on one of the best evaluation algorithms.

Chapter 2

Overview on Genetic Algorithms

2.1 Introduction

Genetic algorithms are a type of evolutionary algorithm that is inspired by nature's creed and is based on genetic science. The main goal of these algorithms is to solve difficult problems that have no exact solution in a reasonable amount of time. These approaches enable evolution from one generation (solution) to the next one. Genetic algorithms are frequently used in many fields to solve optimization or search problems. After presenting the principles and the basic functionalities of a genetic algorithm as well as its characteristics. In this chapter, we will describe the different genetic operations such as mutation and crossover. As well as we will make a comparative study of related works, methods, and metrics used in this field.

2.2 Single-Objective Optimization Techniques

Different optimization techniques have found in the literature that can be divided into three categories (see Figure 2.1) [4].

- Calculus-based techniques(Numerical methods): are further classified into two types (direct and indirect methods). Use a set of necessary and sufficient conditions that must be met by the optimization problem solution. Because calculus-based methods have a local scope and assume the existence of derivatives, they can be very efficient in a small class of unimodal problems.

- Enumerative techniques: entail evaluating every point in the finite, or discretized infinite, search space in order to find the best solution. Dynamic programming is a well-known example of enumerative search.

- Random techniques: Guided random search techniques are based on enumerative methods, but they use additional information about the search space to direct the search to potential search regions. These are further classified into two types based on whether they are searched with a single point or with multiple points at the same time:

– Single-point search: Simulated annealing is a popular example of a single-point search technique that uses thermodynamic evolution to find the lowest-energy states.
– Multiple-point search: evolutionary algorith- ms, such as genetic algorithms (GA), are well-known examples of multiple-point search, in which random choice is used. The guided random search methods are useful in problems with a large, multimodal, and discontinuous search space, and where a near-optimal solution is acceptable.

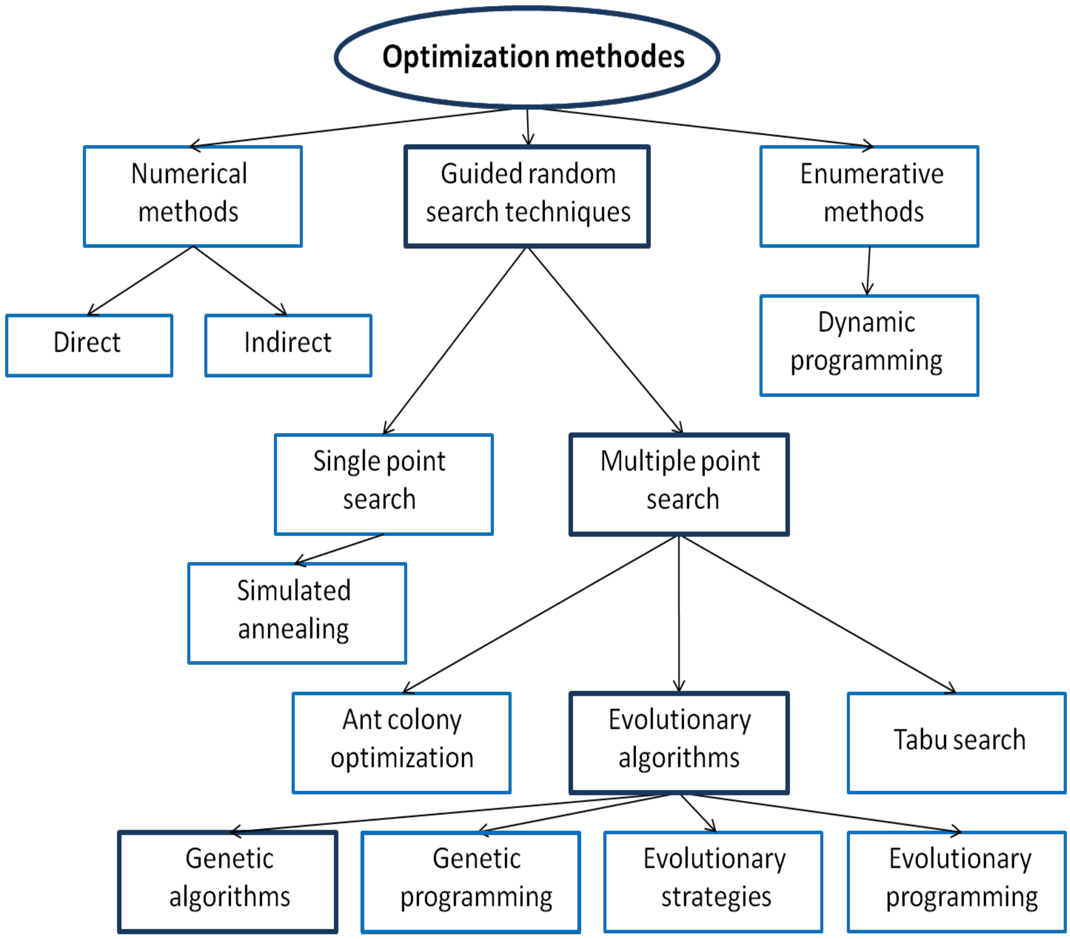

FIGURE 2.1: The different search and optimization techniques [4]

The remainder of this chapter focuses on single-objective optimization methods (multiple-point search), such as evolutionary algorithms in general and genetic algorithms in particular.

2.3 Evolutionary Algorithms (EAs)

Search methods inspired by Darwin's theory of evolution, are stochastic methods that simulate the process of natural evolution in solving optimization problems. which work on a population of potential solutions through iterative phases of selections and random variations. The choice of a representation, the definition of parameters, or the assignment of their own values all have a significant impact on the algorithm's performance. A non-fitness-function-matching option can make the problem more difficult to be [4]. An evolutionary algorithm typically consists of three elements [32]:

- A population is made up of several individuals representing potential solutions to the given problem, allowing for the storage of the results at each step of the search process.
- Mechanism for evaluating (fitness) of individuals allows for the measurement of an individual's quality.
- A population evolution mechanism allows to eliminate of some individuals and create new ones thanks to the predefined operators such as selection, mutation, and crossover. These methods are applicable in most optimization problems (multimodal, non-continuous, constrained, noisy, multi-objective, dynamic, etc).

Four classes of evolutionary algorithms can be distinguished (genetic algorithms, genetic programming, evolutionary programming, and evolutionary strategies), these classes differ only in operator implementation details as well as in population selection and replacement procedures. Although their original purpose was different, they are now primarily used to solve optimization problems (see figure 2.2) [33].

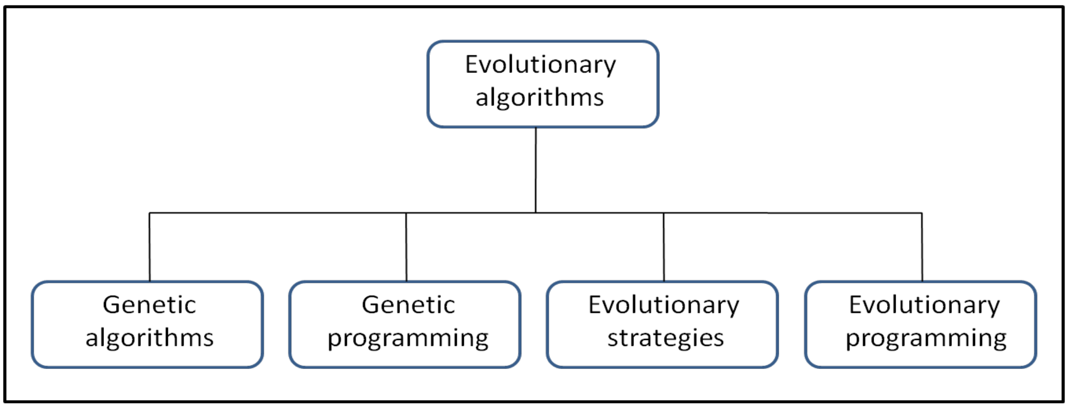

FIGURE 2.2: Classes of evolutionary algorithms

2.4 Overview of Genetic Algorithms

2.4.1 The benefits of the Genetic Algorithms

When compared to traditional optimization algorithms, the genetic algorithm has several advantages, including [34]:

- The only use of objective function evaluation without regard for its nature. Indeed, we do not require any specific property of the function to optimize (continuity, derivability, convexity, etc.), giving it greater flexibility and a wider range of applications.
- The implementation of parallelism forms by working on multiple points at the same time (population size N) rather than just one as in classical algorithms.
- The use of probabilistic transition rules (crossing and mutation probabilities), as opposed to deterministic algorithms, in which the transition between two successive iterations is imposed by algorithm structure and nature. In certain circumstances, This application enables genetic algorithms to avoid local optimums and move towards a global optimum in certain situations.

2.4.2 Nomenclature of the genetic algorithm

As genetic algorithms have roots in both biology and computer science, the terminology used is drawn from both disciplines. The following table defines some terminological [4].

TABLE 2.1: The terminology used in a genetic algorithm

Term	Genetic Algorithm	Biological Significance
Gene	trait, characteristic	a unit of genetic information passed on by an individual to his or her offspring.
Locus	position in the chain GiB	ocation of a gene in its chromosome.
Allele	value of characteristic	one of the different forms that a gene can take, the alleles occupy the same locus.
Chromosome	chain	a structure containing the genes.
Phenotype	set of parameters or a decoded structure	observable physical and physiological aspect of the individual obtained from its genotype.
Epistasis	non - linearity	a term used to define the relations between two "distinct" genes when the presence of a gene prevents the presence of another non-allele gene.

2.4.3 Definition Genetic algorithm

A genetic algorithm is a procedure of research and optimization that is known as an efficient, adaptive, and robust process, which uses random selection as a tool to guide this research in a very large, complex, and multimodal space [4].

GAs are based on natural genetic systems, in which the genetic information of each individual or potential solution is encoded in structures known as chromosomes. The fitness function refers to how they use domain or problem-specific knowledge to direct the research toward more promising areas. Each individual or chromosome has a fitness function that indicates its goodness in relation to the solution.

2.4.4 General principles

A genetic algorithm looks for the extrema(s) of a function defined on a data space. Whatever the problem to be solved, genetic algorithms are based on the following principles [5]:

- A coding principle for the population element: this step assigns a data structure to each point in the state space. It is typically placed after a phase of mathematical modeling of the treated problem, a function that allows the data of the real problem to be modeled in usable data by the genetic algorithm. The type of data coding used influences the success of genetic algorithms [5].

- A mechanism for generating the initial population: this mechanism must be capable of producing a heterogeneous population of individuals to serve as a foundation for future generations. The generation of the initial population is significant since it represents the algorithm's starting point, and its selection influences the speed and optimality of the final solution. In case there is no information about the problem to be solved, the initial population to be distributed across the entire search domain will be critical [5].

- A fitness function: is a function that calculates an individual's fitness level. Which is used to select and reproduce the best individuals from a population. Such that the problem is limited to finding the group of individuals with the optimal values, fitness function guides to the determination of the relevant solution [5].

- Genetic operators: are used for the evolution from one population to another while improving the objective function [34].
  - Crossing operator: it manipulates the structure of the parents' chromosomes in order to produce new individuals.
  - Mutation operator: it avoids establishing uniform populations unable to evolve. It consists in modifying the values of a gene in a chromosome according to the probability of mutation.

- Dimensioning parameters: population size, the total number of generations or termination criterion, application probabilities of crossover, and mutation operators [5].

The general principle of GA function is represented on the figure 2.3.

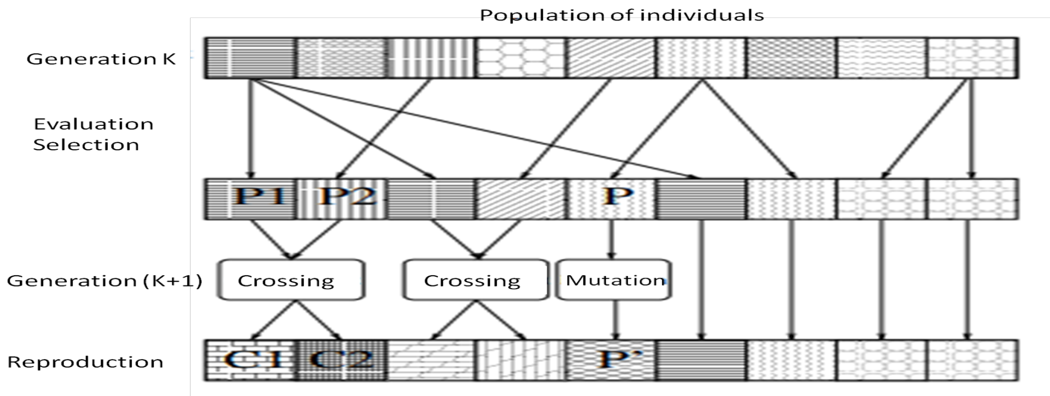

FIGURE 2.3: The general principle of genetic algorithms [5]

2.4.5 Paradigm

In GA, each solution parameter is represented as a gene and all values that can be taken are represented as alleles (**bit**) of this gene. Thus, we have to find a way to code each different allele in a unique way (establish a bijection between the real allele and its code representation). A **chromosome** is a component of the solution that contains a sequence of genes; for example, we can group similar parameters in the same chromosome (single-stranded chromosome), and each gene will be identified by its position: its locus on the chromosome in question. Because each **individual** represents a solution (represented by a set of chromosomes), and a **population** is a collection of individuals, the population is the collection of possible solutions, represented in the figure 2.4.

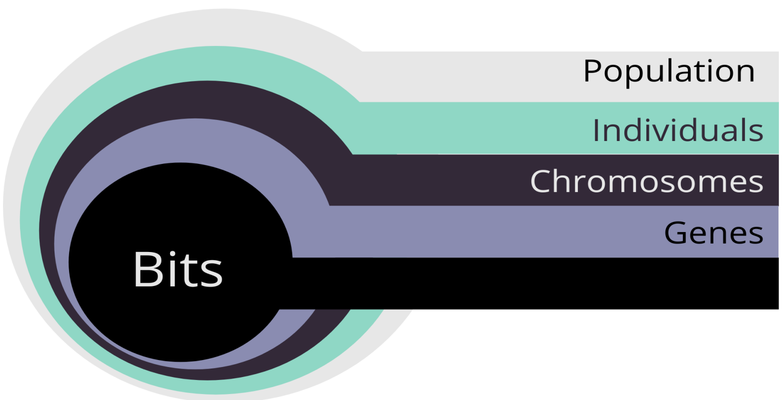

FIGURE 2.4: The five levels of chromosomal representation

2.4.6 Genetic algorithm operators

Genetic algorithms depict a problem-solving approach, presented in the figure 2.5 that begins with the generation of an initial population (chromosomes) of N individuals, for which the values of their objective function are calculated and the individuals are chosen using a selection method. Individuals are chosen with a probability P to be subjects of crossing by the crossing operator. A mutation operator with a mutation probability P can mutate its results. Individuals resulting from these genetic operators will be inserted into a new population using an insertion method, and the value of the objective function of each of its individuals will be evaluated. A stop test will be performed to ensure the quality of the individuals obtained. If this test is successful, the algorithm will be terminated [34].

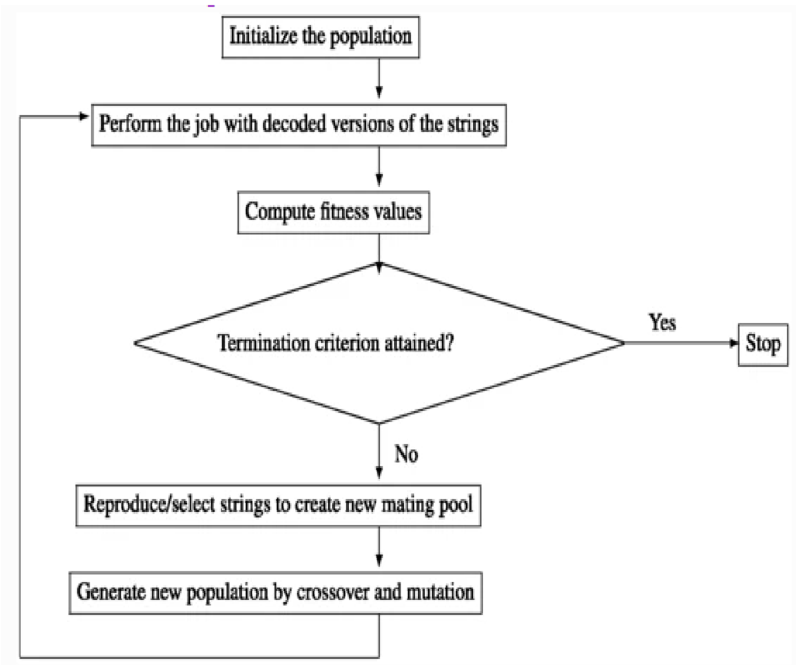

FIGURE 2.5: The genetic algorithm flowchart [4]

Chromosome coding

Genetic algorithms run on a sample of individuals from a population. A chromosome distinguishes these individuals. The coding used by genetic algorithms was represented in the form of bit strings containing all the information necessary to describe a point in the state space, as a result of this mechanism of converting the values of the parameters of a possible solution into strings, resulting in the chromosomal representation [35].

The data coding method used is determined by the specificity of the problem being addressed. It has a significant impact on the genetic algorithm's efficiency.

Depending on the alphabet used, a chromosome (a specific solution) can be coded in a variety of ways.

In general, the most common types of coding are [35]:

- **Binary coding (Digital)** is a type of elementary coding in which the solution is encoded using a string of bits, with each gene having a value of 0 or 1. The traditional data structure is an array, also known as a vector, of Boolean variables.

- **Real or integer coding (Symbolic)** the search space is either an integer or a real space. contrary to binary coding, a gene is represented by a sequence of bits (one bit in the binary code) that is associated with a real. This type of coding is particularly useful when seeking the maximum of a real function, where the variables have defined intervals based on the problem constraints. Each person (solution) is made up of one (or more) vectors of integer or real values.

- **Multiple character coding (Alphanumeric)** in contrast to binary coding, coding the chromosomes in a genetic algorithm using multiple characters is another option. This type of coding is frequently more natural than binary coding.

- **Coding in tree form** this coding employs a tree structure, which is a data structure with a root from which one or more children can be deduced. A tree has two types of nodes: internal nodes (non-terminal symbols) and leaves (terminal symbols). One of their advantages is that they can be used in problems with indefinite solutions.

Population

The algorithm's speed is heavily influenced by the initial population of individuals. If the position of the optimum in the space is completely unknown, it naturally generates random individuals, while ensuring that the individuals produced respect the constraints and are generated uniformly in each associated domain with the components of the solution set. On the other hand, if prior knowledge about the problem is available, individuals will be generated in a specific sub-domain in order to accelerate convergence [5].

Evaluation function

The evaluation function that is also known as performance, adaptation, objective function, or fitness function represents the general step in which each individual's performance is measured. A common evaluation measure must be established in order to compare the obtained individuals with other ones. This method ensures that high-performing employees are retained [36].

Selection

The selection operation determines which individuals are more likely to achieve the best results.

- **Roulette selection** The population is represented as a roulette wheel, with each individual represented by a portion proportional to its fitness value. Individuals are chosen by turning the wheel in front of a fixed pointer. One disadvantage of this type of selection is that the same individual is always chosen even if there are better candidates; this can lead to a loss of population diversity [36].

- **Random selection**The selection is done at random, uniformly, and without using the adaptation value. We are not interested in the fitness value for it each individual has the same probability of being chosen as all other individuals (1 / N). The convergence of the genetic algorithm is generally slow when using this method [36].

- **Tournament selection**This method increases the chances of low-quality individuals in relation to their fitness, i.e. a population of m people. We form m pairs of people. Then, on pairs of individuals, apply the probability of selection proportional to fitness, i.e. the probability of victory of the strongest. This probability represents the likelihood that the best individual from each pair is chosen [36] [34].

- **Elitism**This method of selection allows favors the best individuals in the population. As a result, the most promising individuals will contribute to the improvement of our population. As we can see, this method causes the algorithm to converge prematurely [36].

Crossover

The crossover consists of performing procedures with a certain probability (referred to as the crossover rate [Pc (0,1)]. The primary goal of crossover is to exchange information between selected parent chromosomes by recombining segments of their respective chains [36]. It recombines the genetic material of two-parent chromosomes to produce offspring for the next generation, these children must inherit certain characteristics of their parents. The crossover rate is the percentage of the parent population that a crossover operator will use [4]. There are several types of crossover among which we find:

- **One-point crossover** An operation that allows two people to create new solutions. For each couple, a point of crossing is chosen at random, and the two-terminal subchains of each of the two-parent chromosomes P1 and P2 are exchanged, resulting in two new child chromosomes (see figure 2.6) [36].

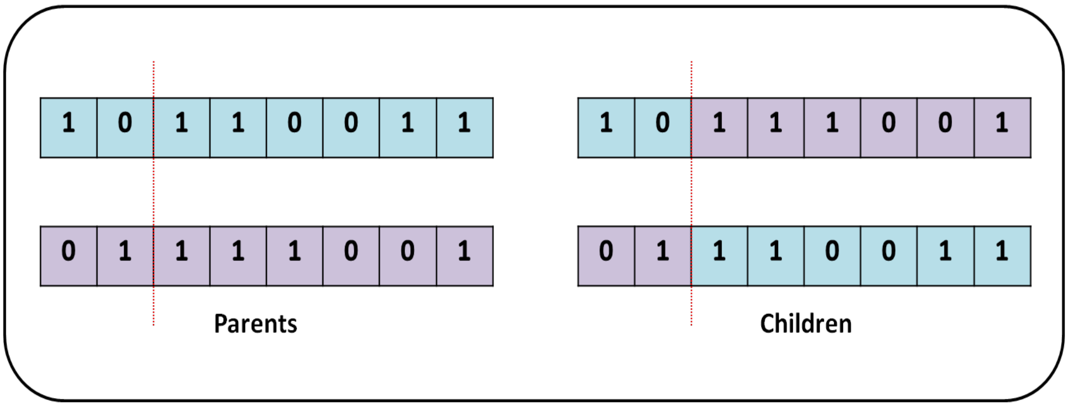

FIGURE 2.6: Crossover at One-point

- **Two-point crossover**At random, two consecutive crossover points are chosen. This operator is generally thought to be more efficient than the one before it [36] (see figure 2.7).

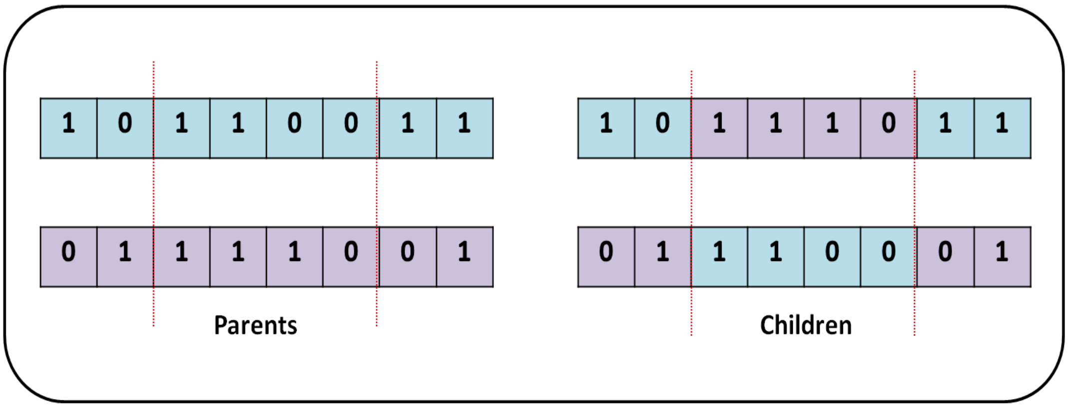

FIGURE 2.7: Crossover at Two-point

Mutation

This operator's role is to randomly modify and introduce genetic diversity into the population. It sometimes allows recovering information lost in previous generations with a very low probability, known as the mutation rate Pm, the value of a population component. In the case of binary coding, the value of an individual component is referred to as the mutation rate Pm. In binary coding, each bit a 01 is replaced with the probability pm [0.001, 0.01] [4]. As a result, the diversity of the child population is preserved. It can affect the fast convergence to a good solution by changing the value of an important bit [36]. Furthermore, it has the potential to slow down the convergence process in the final stage (see Fig. 2.8).

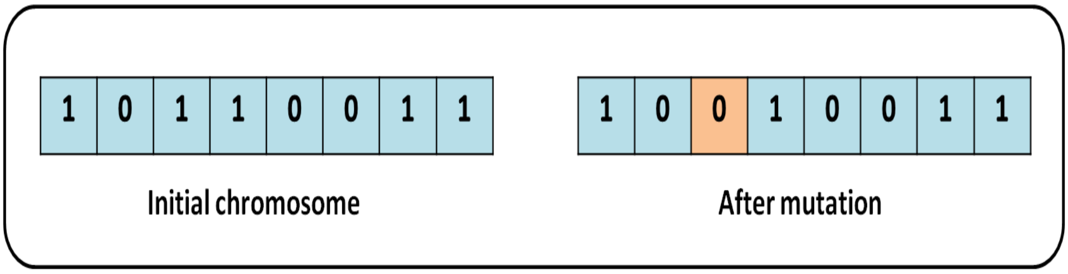

FIGURE 2.8: The Mutation operator

Termination criteria

plays a primordial role in the judgment of the quality of the individuals. Its goal is to ensure that the final solution obtained by the genetic algorithm is optimal. There are two types of stopping criteria [4].

- Stopping after a fixed number of generations. It is the solution that is retained when a maximum calculation time is imposed.
- Stopping when the population stops evolving or does not evolve sufficiently We are then in the presence of a homogeneous population, which we can assume is close to optimal. This stop test remains the most objective and the most used. It should be noted that there is no guarantee of the algorithm's good convergence. The stop is arbitrary, as in any optimization procedure, and the solution "in finite time" is only an approximation of the optimum.

The algorithm below describes a generic version of a genetic algorithm.

Algorithm 1: Pseudo-code of an abstract genetic algorithm

Input: Evaluation function f, maximum number of iterations n
Output: Best individual found
Randomly generate the initial population;
$i \leftarrow 0$;
while $i \leftarrow n$ *Stop condition not met* **do**
    Assess each individual in the population according to f;
    Select parents from the population;
    Produce the children of the selected parents by crossing;
    Mutate the individuals;
    Expand the population by adding the children;
    Reduce the population;
    $i \leftarrow i + 1$;
end
Return the best individual found;

2.4.7 Genetic Algorithm Applications

The last quarter of the twentieth century saw the introduction and rise of natural-source optimization techniques such as genetic algorithms (GAs) (Holland 1975; Goldberg 1989), simulated annealing (Kirkpatrick et al. 1983), and ant colony optimization (Dorigo 1992). These techniques are being used to solve engineering problems where traditional optimization methods are insufficient.

It is well known that construction scheduling problems with unlimited resources (CS/UR) can be solved by using deterministic techniques such as the critical path method (CPM) and linear programming (LP) if the duration of the activities is predefined. But stochastic methods are to be used in cases where durations are to be determined or other parameters such as multi-modal activities, resource leveling, and cost minimization are tackled. Genetic algorithms constitute one such method.

- They have been applied to scheduling problems in the last few decades. In a survey about CS/RC problems made in 1998 (Herroelen et al. 1998).
- in a similar survey made in 1999, they take a minimal space (Brucker et al. 1999).
- Basic information about the application of GAs to project schedules can be found in Wall (1996).
- Al-Tabtabai and Alex (1998), and (Hegazy 1999) applied GAs to construction problems for crash schedule determination or time–cost trade-off scheduling problems. Improved heuristics were proposed and used with GAs for applications to resource allocation and leveling problems (Hegazy 1999).

2.5 Related Work

Several research papers have been published in recent years that address the node deployment problem in order to achieve maximum coverage and connectivity in WSN. The goal is to find the optimal placement of nodes so that only a few numbers of them are required. The main challenge is to find many near-optimal solutions (optimal deployment) within a limited computing time.

The following work implements optimization of node deployment in different areas trying to improve some features such as cost, energy consumption, coverage, connectivity, number of nodes used, etc. Among the existing work, we have chosen the following:

In 2021, the authors in [37].have proposed a repairing a coverage hole Mechanism that can take into account the simultaneous death of multiple sensor nodes and the area coverage repair method (Genetic Algorithm based Coverage hole Repair mechanism, GACR). While GACR uses the genetic algorithm to divide the dead nodes into groups and uses the neighbors of the dead node as the repair node for each group. The simulation results show that GACR maintains more than 90 % of network coverage time. Compared with the previous literature that mentioned NIFP and DCHR.

M. Choudhary and N. Goyal, 2022. A dynamic topology control algorithm for node deployment (DTCND) to address the drift of underwater wireless sensor networks, is suggested in this article [38]. In order to ensure coverage and connectivity, this work aims to monitor node mobility and predict nodes' locations. In order to meet the coverage and connectivity requirements. At various depths, the sensor nodes are randomly deployed. Simulation results demonstrate that the suggested algorithm achieves 5 %, and there is a 9 percent increase in connectivity, the residual energy is higher by 3 and 18 percent, respectively.

The authors in this document [39], discussed the pursuit of better sensor node locations in order to generate a maximum number of disjointed sets. The ultimate objective is to prolong the target monitoring network's lifespan. This work examines the nature of the problem and proposes a multi-objective genetic algorithm by means of the NSGAII approach that simultaneously seeks out the best sensor placements and the greatest number of disjoint coverage sets (MDSC).

In 2020, this article [40], B. Foubert and N. Mitchell. This paper proposes simplifications for the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). In order to choose the best communication interface, The deployment of a network composed of systems that are typically energy-constrained and communicating via wireless links is constrained by the radio range and speed of the technology used. According to them, the evaluation of the solution using Pycom FiPy modules reveals a 40 % improvement in calculation speed while guaranteeing an 81 % ranking similarity with TOPSIS.

Following our review of the related work presented, in this section, we conducted a comparative study of this work (The following table) with respect to the main deployment objectives discussed in the literature (metrics used for optimization), optimization methods used, and type of deployment.

In the last line of the table, we give the information related to our proposed solution (see next chapter). As outlined in the table, we propose a solution that maximizes target coverage, ensures connectivity, minimizes overlap, reduces the number of active sensor nodes deployed in the network, and helps extend the network's lifespan.

TABLE 2.2: Comparison of related work

Application	Metrics	Methods Used	Deploymen	Ultimate goal
Coverage Hole Repairment for Mobile Wireless Sensor Networks with Simultaneous Multiple Node Deaths [37]	Coverage	GACR	Not mentioned	Coverage Hole Repairment for Mobile Wireless Sensor Networks
Dynamic topology control algorithm for node deployment in mobile underwater wireless sensor networks [38]	Coverage, connectivity, Energy-efficient	DTCND	Randomly	Detect node disconnections in mobile underwater wireless sensor networks
Evolutionary Approach to Deployment of Wireless Sensors for Target Coverage [39]	Coverage	NSGAII	Randomly	Cover completely, Extend the life of the network
Communication interface selection in multi-technology sensor networks [40]	Communication interface, Calculation time	TOPSIS	Not mentioned	Selection of communication interface
Discussion:	Coverage, Connectivity, Number of Active Nodes, Energie	Genetic Algorithm	Randomly	Find the optimal deployment of WSN

2.6 Conclusion

In this chapter, we defined the single-objective optimization problem and presented the various approaches that have been proposed to solve these problems. Where we laid the groundwork for understanding genetic algorithms, which will be used in designing the proposed system. We detailed the various steps that comprise the general structure of a genetic algorithm and conducted a comparative study of related works, methods, and metrics used and we have indicated our goal and discussion. The wireless Sensor Network Deployment problem formulation is presented in the following section, and we used Algorithm Genetic to solve it and get an optimal deployment, which describes the formulation of GA and the parameter choices used. This is followed by a section in which the obtained results are discussed and conclusions are drawn.

Chapter 3

Temperature-Aware Dynamic WSN Deployment

3.1 Introduction

This chapter presents the global architecture of genetic algorithms that are considered one of the most efficient optimization techniques, also the modifications performed on it tend to reach the near-optimal solution. After that, it will clarify the requirements analysis and the user system design. A the end, a genetic algorithm will be applied in order to optimize WSN deployment under temperature variations.

3.2 General context

Figure 3.1 represents the global architecture of WSN in our system. The main objective of this research project is to optimally deploy several nodes in an equipped area to build a sensor network capable of detecting information such as temperature and humidity in real-time. The structure of the area is taken into consideration in the process of randomly deployed sensor nodes in a 3D area, alongside the coverage, connectivity, number of nodes, and overlap between sensor nodes. These sensors are connected to a sink node (base station node) that provides two functionalities:

- Run the Genetic Algorithm on the initial sensor nodes deployment.
- Broadcast the near-optimal deployment as a reconfigurable matrix (RMatrix).

In this work, we propose a deployment system based on a genetic algorithm (GA). The deployed sensor nodes must be connected to ensure they are ready to communicate with the sink node. Every node shall send its position to the sink node to identify the near-optimal deployment using a specific fitness function. Once the sink node identifies the near-optimal deployment, it has to broadcast this new deployment as RMatrix.

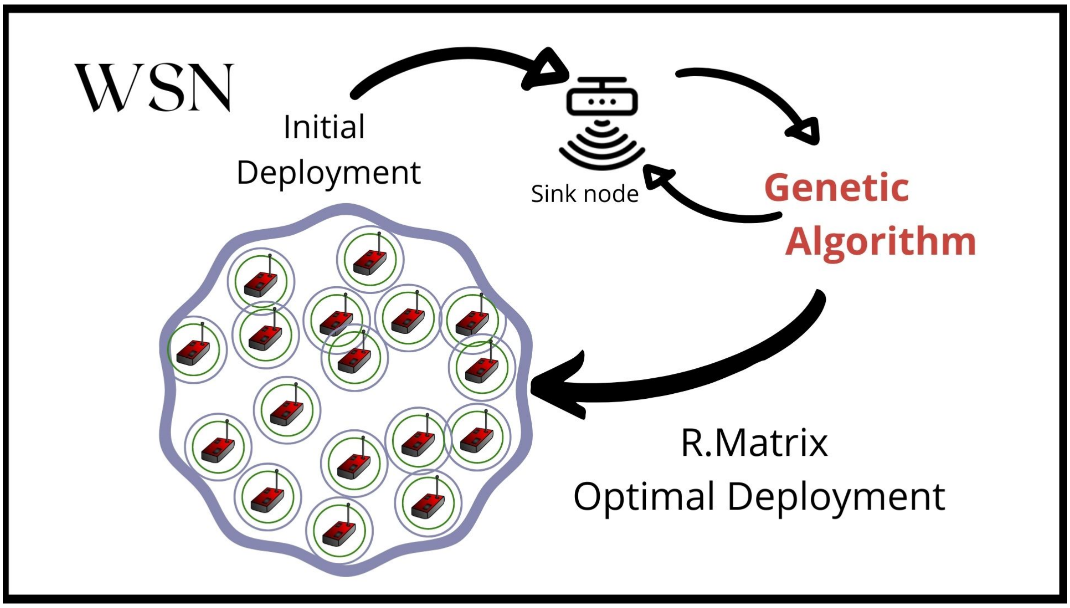

FIGURE 3.1: Global Scheme

3.3 WSN deployment issue

The near-optimal deployment of WSN is a critical issue that has to be adaptable to environmental changes. In order to identify this deployment, we attempt to use the concept of biological chromosomes and genes (i.e., GA). GA is the most widely used meta-heuristics method to solve stochastic optimization problems. Starting from a random search with no prior information, the genetic operators guide the evolution of chromosomes which represent the solution set in this case, to the optimum solution. The problem statement can be described as follows:

- Let A be a 3-dimensional space, in which a WSN is going to be deployed;
- The WSN deployment is defined by sensor nodes coverage and communication range;
- To ensure network connectivity and coverage, we have to best positing of each sensor node on this space A;
- In order to avoid redundantly in network connectivity and coverage, we have to define the necessary number of sensor nodes that ensure network requirements under environmental changes such as temperature variations.

3.4 System design

The optimization is carried out by our system using an evolutionary algorithm. In this section, we'll show how to adapt the Genetic algorithm to this problem (WSN deployment optimization) by following GA steps until identify the near-optimal solution. Then, we'll go over describing each part of the GA in detail.

3.4.1 WSN Deployment

In order to cover hostile areas, sensor nodes are dispersed randomly in the air by an aircraft or drone. Which needs thousands of hundreds of sensor nodes to ensure network connectivity and coverage. However, network connectivity and coverage cannot be guaranteed due to the lightweight of sensor nodes, which gave them a high air resistance. The generated distribution is called simple scattering [25].

3.4.2 Communication Range

The quality of a wireless communication link between two nodes, a transmitter and a receiver, is usually expressed by the experienced Packet Error Rate (PER). Where the communication range is defined as the distance over which less than 1% out of packets are lost (corresponding to less than 1% out of PER) [8].

Effect of Temperature

Bannister and colleagues demonstrated that high temperatures impede communication between sensor nodes. The temperature has an impact on both transmission and reception quality. The effect of temperature on transmission power and reception sensitivity is investigated, and the values of sensitivity Power (P_s) and P_t (Transmission Power Radiated by the transmitter) are found to vary with temperature. As a result, the actual value of communication Range (R_c) is temperature-dependent [8].

3.4.3 On Connectivity and Coverage

Connectivity Criteria

Connectivity is considered a critical part when it comes to transmitting data to the base station and performing necessary actions. A WSN is considered connected if at least one path exists between the sink and each sensor node in the area, and each node must be able to communicate with at least one other node. Connectivity in a network is a measure of fault tolerance or path diversity. The 1-connection is a necessary condition for the network to be operational. The maximum distance between nodes that can communicate with each other is defined as the communication range (R_c). In another way, the probability $P(n_i, n_j)$ that node n_i is connected to node n_j is defined as follows [24]:

$$P(n_i, n_j) = \begin{cases} 1 & if \quad d(n_i, n_j) \leq R_c \\ 0 & Otherwise \end{cases} \tag{3.1}$$

Where $d(n_i, n_j)$ is the Euclidean distance between n_i and n_j. This distance is given by:

$$d(n_i, n_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3.2}$$

For connectivity, $Con(n_j)$ indicates the set of sensor nodes that are within the connectivity range R_c of sensor node n_i defined by:

$$con(n_i) = n_j / d(n_i, n_j) \leq R_c, \forall j, 1 \leq j \leq N \tag{3.3}$$

The connectivity of a network is expressed as follows :

$$\mu(R) = \frac{N \times \pi \times R_c^2}{A} \tag{3.4}$$

Coverage Criteria

Coverage is another important factor in deploying a WSN. Its goal is to ensure that every target in the surveillance area is covered by at least one sensor node. In order to ensure network reliability, the deployed sensor nodes must provide complete coverage of the events occurring in the deployment space. The total area covered by a set of deployed sensor nodes in the region of interest is expressed as the event coverage capability.

In other words, a point p in the deployment space is said to be covered by a node n_i if and only if p is within the node n_i perimeter and there is no obstacle between p and n_i.

Considering the communication range equation that is known by :

$$R_c = \sqrt{3} \times R_s \tag{3.5}$$

we can use it to calculate Rs that's defined as :

$$Rs = \frac{R_c}{\sqrt{3}} \tag{3.6}$$

3.4.4 Energy Constraints

In physical mobility, then the energy is greatly dependent on the mechanics that "move" the sensor or the node. Based on the energy conservation principle. [41]

- In the case of frictionless motion: the exerted energy to accelerate an object to a speed (v) should equal the kinetic energy that the object gains:

$$E = 1/2mv^2 \tag{3.7}$$

- In the case of constant speed sensors, modeling the mobility energy as the following:

$$E_m = a * d + c \tag{3.8}$$

Where:

  - d is the traveled distance in meters.
  - a models the continuous friction losses in Joule per meter.
  - c is the required kinetic energy.

3.5 Objective and hypothesis of the work

3.5.1 Objective

The objective of the proposed work is to find the near-optimal deployment of WSN in a 3D space with a minimum number of sensor nodes. In this thesis, we apply a Genetic Algorithm to optimize WSN connectivity and coverage.

3.5.2 hypothesis

In order to solve this optimization problem and specify the characteristics of the space on which our solution is applicable. We have defined a set of assumptions as follows:

- The deployment space is defined by a 2D raster representation which is $(X, Y, Z = 0)$.
- Random initial deployment in 2D space expresses the location of each node n with (Xn, Yn).
- The random deployment is generated by moving nodes randomly in 2D space.
- Inside each node, there are wireless sensors (such as temperature sensors) that detect and collect information about the environment.
- The used sensor nodes are mobile.
- Network deployment according to temperature variations is renewed every K time.
- The sink node is responsible for collecting the initial positions of sensor nodes, after that applying the GA in order to obtain the near-optimal position of each node. In the end, once the near-optimal deployment is defined, the sink node has to broadcast a reconfigurable matrix (RMatrix) containing the new position of each node. This process is repeated every k time according to temperature variations.

- The sink node has to memorize a copy of the last obtained RMatrix in order to use it as initial deployment for the next round (after k time).

3.6 Detailed architecture

In order to solve the deployment problem, we proposed a new approach based on mono objective optimization to the adaptation of the objective optimization method (GA) that was presented previously. The objective of the proposed system architecture is defined in Figure 3.2.

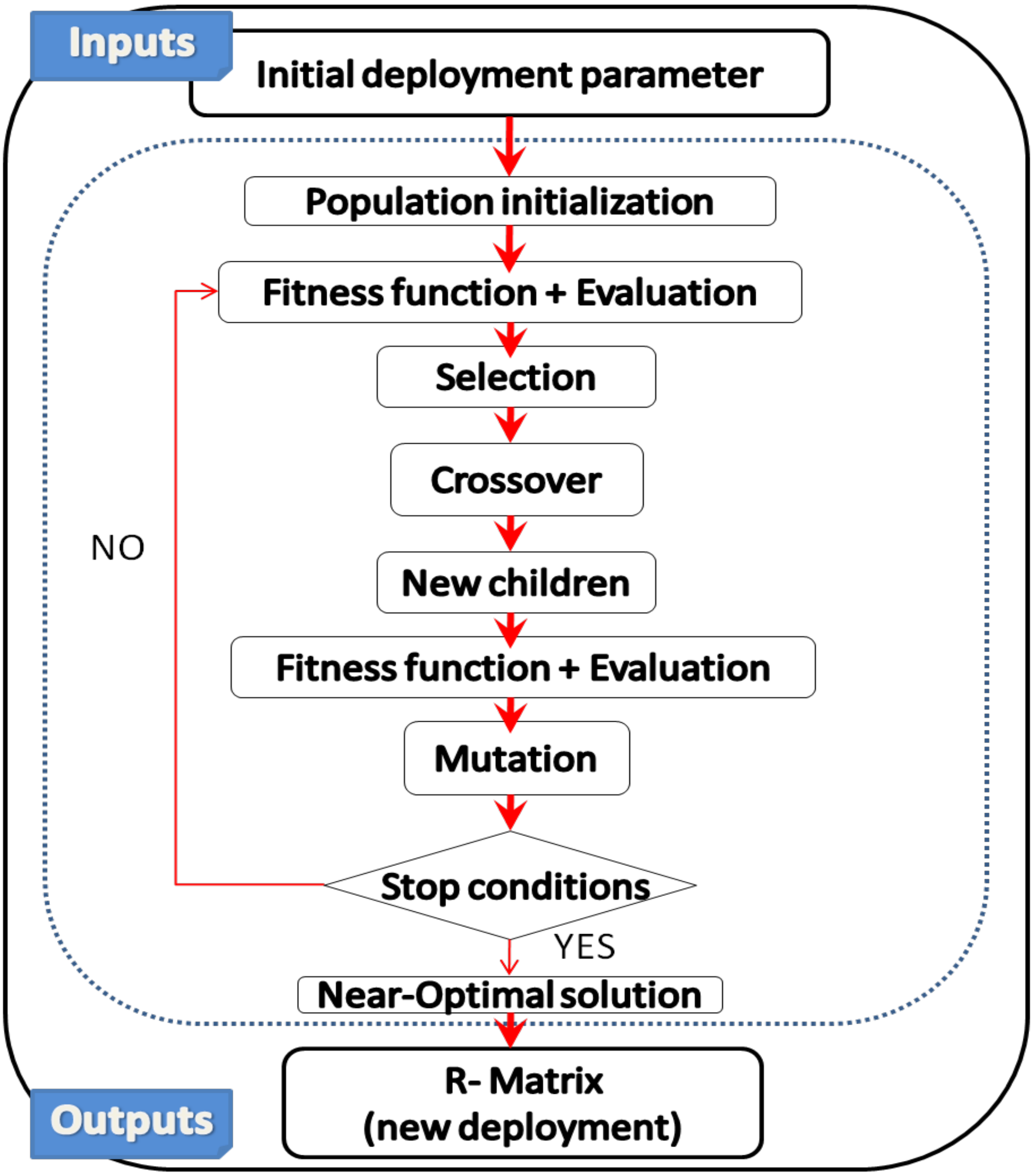

FIGURE 3.2: Detailed system architecture

3.7 UML Sequence

After the initial deployment of sensor nodes, they have to send their locations to the sink node. Once the conditions (temperature changed + time) are verified, the sink node has to apply GA at least one time. In order to select the best chromosome among the population (set of random deployment), the fitness of the whole population has to be calculated. Once the optimal chromosome is identified, the sink node has to broadcast it as an RMatrix if and only if the obtained deployment is totally different from the old one (see figure 3.3).

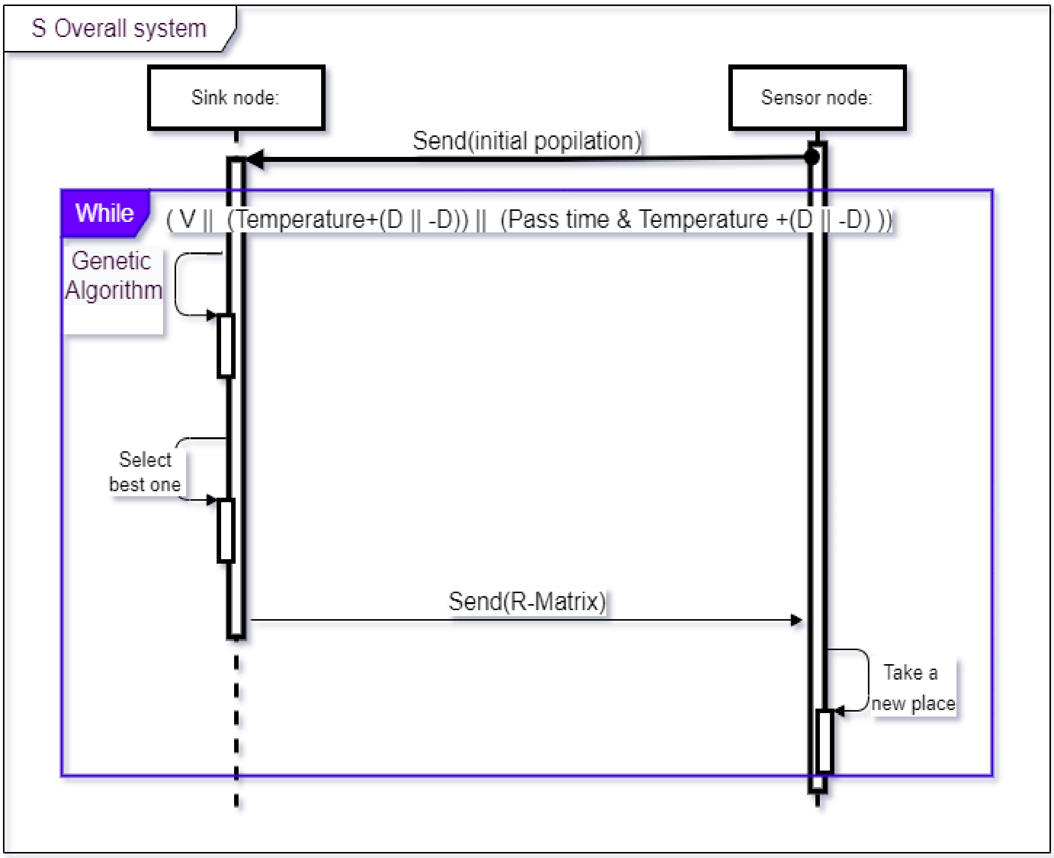

FIGURE 3.3: UML Sequence

3.8 Formalization model

In this section, we present a mathematical formalization of the main purpose using linear programming, as follows in the table:

Objective function

- Minimize $F = N - Ni$
- Maximize $F' = d(n_i, n_j) - R_c$

In order to meet the objectives imposed in the proposed system (i.e., maximize coverage and connectivity, minimize the number of active nodes), We attempt to minimize the total number of active nodes and maximize the distance between each node and its neighbors to be equal to the communication range.

TABLE 3.1: Variable choices and Constraints

Variable	Abbreviation and Constraints	Variable	Abbreviation
E:	Energy >0	T:	Temperature.
R_c:	Communication Range>0	N:	Number of nodes.
$d(n_i, n_j)$:	Distance between node $i, j \leq R_c$.	N_i:	Number of nodes inactive.

3.9 The proposed solution

In order to obtain a fully covered and connected WSN with the minimum number of sensor nodes, we propose to apply the genetic algorithm on the WSN random deployment (initial publication) for defining and then identifying the near-optimal one. the following steps explain how this near-optimal solution will be identified.

3.9.1 Initial population

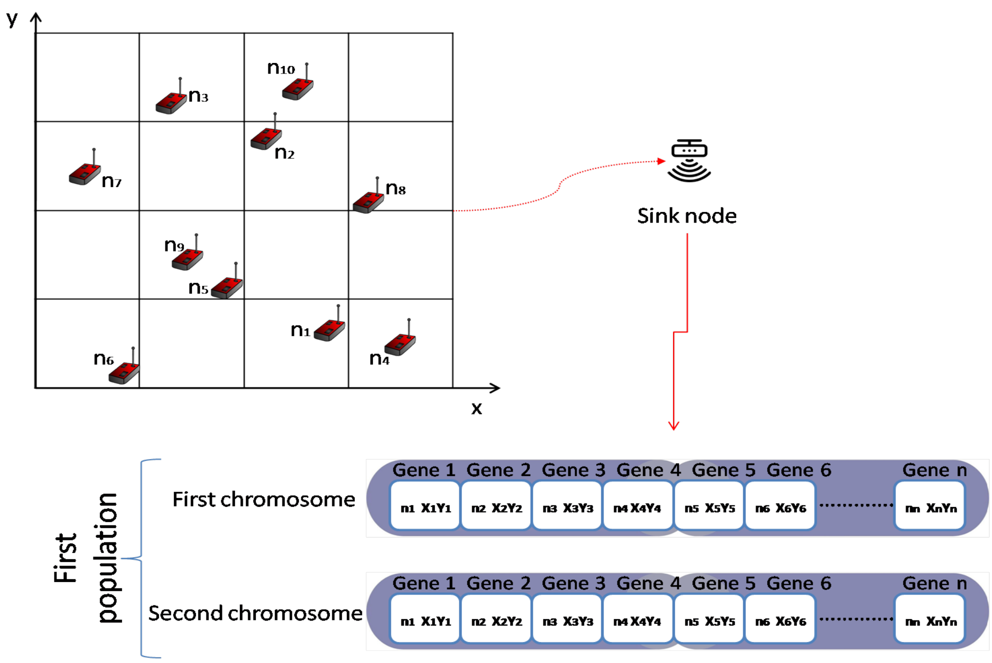

FIGURE 3.4: Initial population

The initial population (see figure 3.4), according to the initial deployment is defined as a set of solutions established by the sink node which is presented in the form of chromosomes. The first solution is the first randomly generated deployment in the grid. The other solutions presents a random positions of sensor nodes (coordinates) to X_n, Y_n.

Randomly generating individuals allows for creating a diverse population while keeping only feasible solutions. This process is presented in the algorithm pseudo 2.

Algorithm 2: Initial population

Input: Nodes, nbrIteration
`/* Chrom[i][j]:coordinates X,Y of node j of chrom i */`
Output: Chrom[i][j]
`/* Initial solutions established like a chromosomes */`
for $i \leftarrow 0$ *to Nbrnodes* **do**
  for $j \leftarrow 0$ *to sizechrom* **do**
    if *First chromosome* **then**
      Chrom[i][j] = N_j ;
    else
      `/* other chromosome. */`
      Chrom[i][j] = random coordinates ;
    end
  end
end

3.9.2 Fitness Function

Algorithm 3: Fitness

Input: R_c,C_{ij}
/* y_{ij}:fitness, C_{ij} : $chrom[i][j]$ */
Output: y_{ij}
for $i \leftarrow 0$ *to Nbrnodes* **do**
  for $j \leftarrow 0$ *to Nbrnodes* **do**
    $d(n_i, n_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$;
    $y_{ij} = d(n_i, n_j) - R_c$;
  end
end

Having the best nodes deployment that provides perfect coverage of the entire space is considered an overall goal. Therefore, we must ensure that the collected data is properly routed to the sink by maintaining network connectivity.

The fitness value for each chromosome (solution) is derived from the system's objectives. To maximize coverage, and connectivity and minimize the number of active nodes, we derive a single fitness function that aims to increase the distance between the two nodes ni and nj with a condition to do not surpass R_c.

$$F = d(n_i, n_j) - R_c \tag{3.9}$$

3.9.3 Selection Criteria

The number of individuals who participated in the reproduction is determined by the selection phase. Individuals with a higher fitness value than others are frequently chosen. The selected individuals are known as parents, who they are chosen to participate in the next phase of reproduction. In our case, the participant's parents are selected based on their fitness value. The parent with the highest and second-highest fitness values are selected for the next GA step (crossover step).

3.9.4 Crossover

In general, crossover entails a process of crossing previously selected parents in order to produce new ones. In this function, the crossover is performed at X point in order to generate new children (individuals). At a crossover point, chromosomes crossed over each other to produce two new children (two new chromosomes), as described in the following figure 3.5 and Algorithm 4:

Algorithm 4: Crossover

```
/* Chrom[p1][j],Chrom[p2][j]:parents                                  */
Input: Chrom[p1][j],Chrom[p2][j], offspring
/* Chrom[i][j]:coordinates X,Y of node j of chrom i                   */
Output: Chrom[i][j],Chrom[i+1][j]
for j ← 0 to offspring do
    Chrom[i][j]=Chrom[p1][j];
    Chrom[i+1][j]=Chrom[p2][j];
end
for j ← offspring to Nbrnodes do
    Chrom[i][j]=Chrom[p2][j];
    Chrom[i+1][j]=Chrom[p1][j];
end
Add( Chrom[i][j],Chrom[i+1][j]);
/* Add: function adds the tow chromosomes in the population           */
```

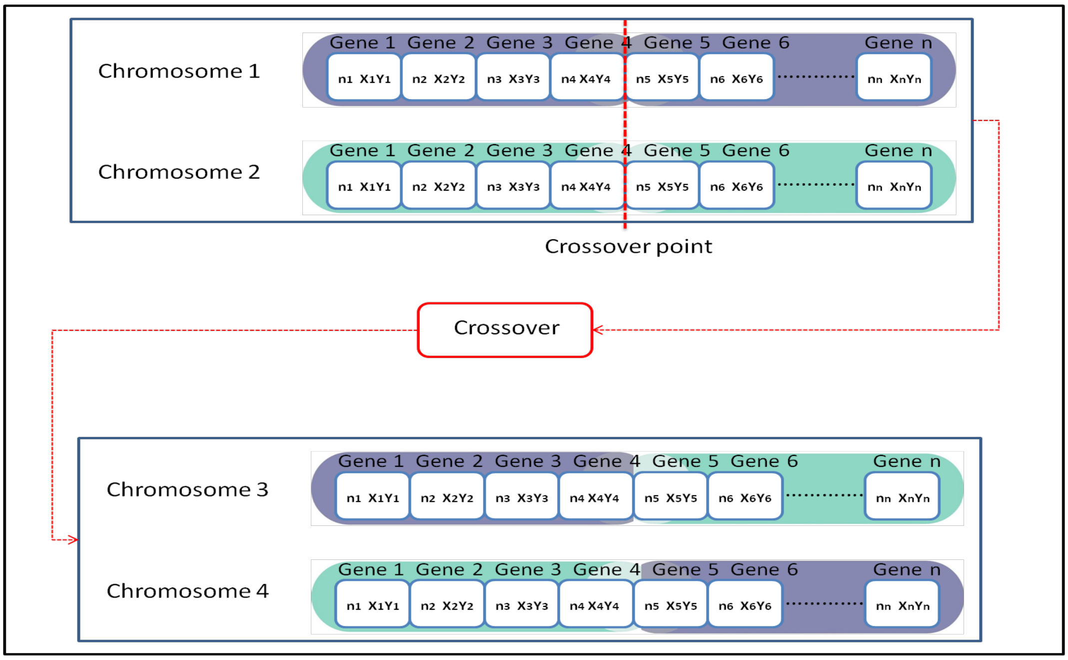

FIGURE 3.5: Crossover function

Constraints

- Select two individuals.
- Determine the X points for crossover function.
- Cross over the selected parents to generate two children.
- Integrate the obtained children into the global population.

Notes: The children must inherit all of their parents' genetic characteristics.

3.9.5 Mutation

Algorithm 5: Mutation

```
/* Chrom[p1][j] have best fitness */
Input: Chrom[p1][j],Chrom[p2][j]
Output: Chrom[p1][j]
for j ← 0 to sizechrom do
    Chrom[1][j].minfit=Chrom[p2][j];
    /* Chrom[1][j].minfit:smallest fitness value in chrome */
end
```

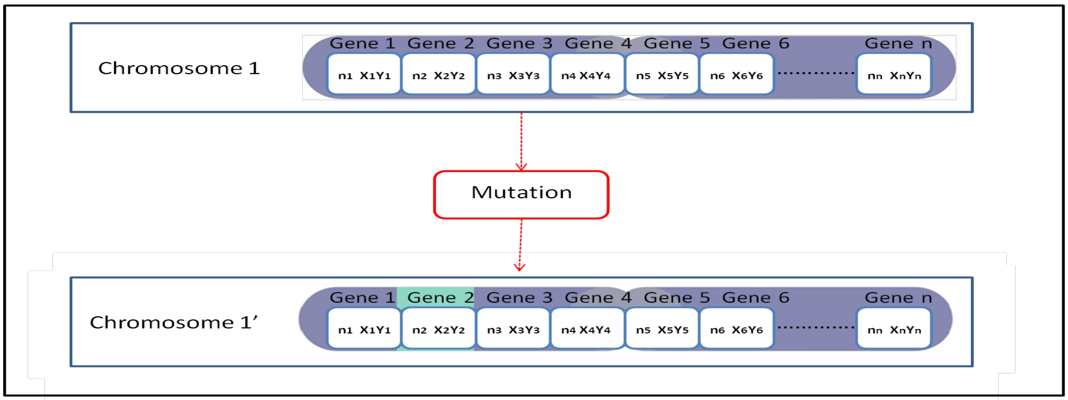

FIGURE 3.6: Mutation

Once the crossover phase is completed, a genetic mutation factor is used for increasing population diversity and providing more candidates. Firstly, we select the chromosome with the biggest fitness, then we look for the gene (the allele) with the smallest fitness value and replace it with another one (replace the coordinates of the node located in this chromosome with other coordinates). The mutation operator enables early convergence to near-optimal deployment, see below Figure 3.6 and Algorithm 5.

3.9.6 Termination Criteria

As long as the stop criterion, which is a probability of coverage and connectivity value between 1 and 0.90, as defined in Eq. 3.1 and Eq. 3.8 is not met, the algorithm continues to operate, and at each temperature variations, the system is reset to zero (see Fig. 3.2).

3.9.7 Near-Optimal Solution

The near-optimal solution is the one with the best network deployment that ensures connectivity and coverage with the fewest number of sensor nodes. In order to save energy, the remaining sensor nodes are turned into hibernate mode (sleeping mode) after fulfilling the termination criteria condition.

3.10 Conclusion

In this chapter, we presented a dynamic WSN deployment using a genetic algorithm under temperature variations. Where the deployment that ensures network connectivity and coverage with the fewest number of active nodes is selected as the near-optimal solution.

We noted that at the outset a system architecture can lead to a suitable design to optimize the WSN deployment. Our goal was to maximize sensing area connectivity while minimizing overlap and the number of active nodes deployed while ensuring network coverage.

In the following chapter, we will describe the simulation environment and evaluate the performance of our deployment system with final results, as well as detail the simulation of this design and the tools used to implement the proposed system.

Chapter 4

Experimental results and Discussion

4.1 Introduction

Network simulators offer a lot of savings, time, and money for the completion of simulation tasks and are also used so that network designers can test new protocols or modify the existing protocols in a controlled and productive manner. Simulation allows developers to discover potential limitations, problems, and design errors to correct them.

After introducing the system's architecture in the previous chapter. In this chapter, we will describe the various technical aspects related to the implementation and deployment. Starting with the evaluation of the proposed model in the context of a large-scale deployment; to demonstrate the efficiency and applicability of the approach in realistic scenarios. After that, we present and discuss the theoretical part of the proposed project, then detail the used approaches and algorithms. Firstly, we provide brief definitions of the development tools used for system implementation, such as the simulation environment and programming language. The implementation details are then presented. Finally, we present the system's results which are supported by experimental results from a comparative analysis.

4.2 Environments and development tools

4.2.1 Experimentation and simulation environment

TABLE 4.1: Characteristics of the machine

CPU	Intel® Core™ i7-4790 CPU @ 3.60GHz
Hearts	8
Memory	15.5 GiB
OS	Ubuntu 20.04.4 LTS

Experimentation and performance simulation of our deployment system is performed using the simulation environment Qt Creator and the network simulator

NS3, which are configured and performed on a desktop computer with the following characteristics:

4.2.2 Programming Language

The environment we have chosen in our study is "C++".There are many programming languages, and C++ is one of the most popular in the world. C++ is a powerful, portable, and open-source programming language with no proprietary components. It is a cross-platform language that anyone can use to create powerful applications.

C++ has a long history, dating back to 1979 when it was created by Bjarne Stroustrup as an extension to the C language [42].

C++ is an object-oriented programming language (it supports classes and objects), which gives programs a clear structure and allows for code reuse, lowering development costs.

Additionally, it is used to create applications that can be adapted to multiple platforms [42].

FIGURE 4.1: C++ icon

4.2.3 Environment description

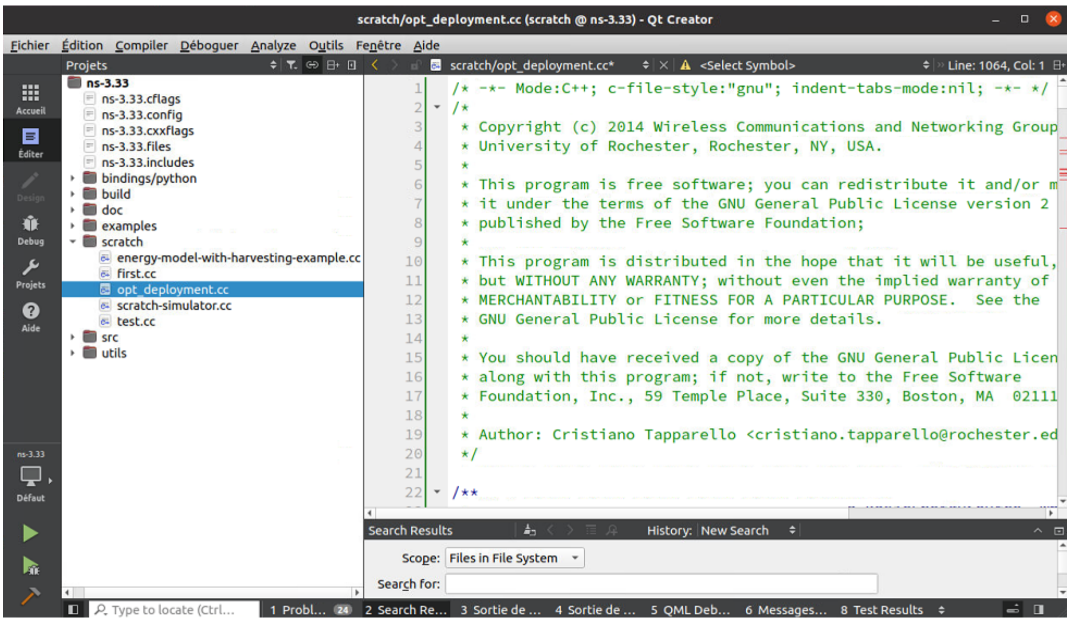

FIGURE 4.2: Work environment

4.3 Choice of simulator

4.3.1 NS3

In order to get our work done, we used Qt Creator which is a cross-platform integrated development environment (IDE) and an advanced code editor built for maximum developer experience. It offers us writing software in C++, QML, JavaScript, Python, and other languages. It features code completion, syntax highlighting, and refactoring and has built-in documentation at fingertips (See figure 4.2) [43].

Among the many WSN simulators, NS-2 and NS-3 are the most popular ones for simulating wireless communication protocols. Since it was necessary to choose only one simulator and NS-3 is not backward compatible with NS-2. NS-2 development started in July 2006. After that, the developers of NS-3 decided that the simulation architecture should be completely redesigned from the scratch files, with keep in mind, the gained experience from NS-2. The first major public and stable version were released in June 2008. It is used on Linux/Unix, OS X (Mac) [44].

NS-3 frees the constraints inherited from NS-2 and allows the construction of a simulator that is well designed from the beginning. Although the number of models and codes provided by NS-3 is limited compared to NS-2, NS-3 performs better than NS-2 in :

- Bi-language (C + C++ / TCL) makes the system complex in NS-2, but for NS-3 everything is written in C++ or Python (only language architecture is more robust in long run).
- NS-3 is faster in terms of execution compared to NS-2 (everything is pre-compiled).
- In terms of memory management, NS-3 is more efficient than NS-2.
- NS-3 has an emulation mode, which allows integration with real networks.

4.3.2 NetAnim

NetAnim is a standalone offline animator built on top of the Qt toolkit (object-oriented API developed in C++). It currently uses an XML trace file collected during the simulation to animate the simulation. this trace file is generated by an animation interface included in the current versions of NS-3. The first version was developed by George F Riley [45].

4.4 Implementation of the Proposed Approach

4.4.1 Libraries used

As a first step, we start importing a different set of libraries, each one of them has a different purpose as presented below (Fig.4.3 and Fig.4.4):

- **Necessary libraries for NS-3.**

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/mobility-module.h"
#include "ns3/config-store-module.h"
#include "ns3/wifi-module.h"
#include "ns3/internet-module.h"
#include "ns3/applications-module.h"
#include "ns3/energy-module.h"
#include "ns3/propagation-loss-model.h"
#include "ns3/propagation-delay-model.h"
#include "ns3/error-rate-model.h"
#include "ns3/yans-error-rate-model.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/wifi-radio-energy-model-helper.h"

#include "ns3/ptr.h"
#include "ns3/animation-interface.h"
#include "ns3/netanim-module.h"
#include "ns3/gnuplot.h"
#include "ns3/object.h"
#include "ns3/nstime.h"
```

FIGURE 4.3: NS-3 Libraries

- **Libraries used for the C++.**

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <sstream>
#include <stdio.h>
#include <stdlib.h>
#include <cstdlib>
#include <ctime>
#include <iomanip>
#include <math.h>
#include <cmath>
#include <vector>
#include <list>
#include <random>
#include<time.h>
```

FIGURE 4.4: C++ Libraries

4.4.2 Conflict resolution by Genetic Algorithm

After setup the working environment to be used, we will see in this section the proposed algorithm code to develop the most important procedures functions used in the GA version.

Initial population

- The population is represented as a structure of chromosome tables, its size will be defined by arithmetic calculation.
- The chromosome is represented by a table of type genes, each gene in the table consists of coordinates (node name, X, Y, Z), see figure 4.5.

```
for(int i=0;i<nbrIteration;i++ )
    for(int j=0;j<chromsize;j++ )
    {
        if (i==0)
         { Vector v= GetPosition(nodes.Get(j));
             chrom[i][j].i=j;
             chrom[i][j].x=v.x;
             chrom[i][j].y=v.y;
         }
         else
         {
             chrom[i][j].i=j;
             Rc=rand()%200;
             chrom[i][j].x=Rc;
             Rc=rand()%200;
             chrom[i][j].y=Rc;
         }
    }
}
```

FIGURE 4.5: Population creation code

Fitness Function

The evaluation of chromosomes is done using a fitness function. Where for each generation step, the following calculations are performed, see figure 4.6:

```
double y=Distance(ch,i,j)-R;
if (i!=j && (y<=0.))
 {
    Neighbor[i][k].v=j;
    Neighbor[i][k].f=y;

    if(y>-1.){
    sumFit++;
    }
     k++;
 }
```

FIGURE 4.6: Evaluation by fitness function code

- Calculation of the evaluation function fitness[i] for each chromosome i.
- Add the fitness of new offspring in the previous fitness table.

Selection Criteria

In order to select the best chromosomes, at each time, we select the two best chromosomes' fitness. After that, the selected chromosomes will pass by the crossover step in order to generate new ones. Afterward, the best one of them will be selected for the genetic mutation step, see figure 4.7.

```
for(i=0;i<Nbchrom;i++)
{
    if( max1<fitness[i])
    {
        max2=max1;
        max1=fitness[i];
        best[1]=best[0];
        best[0]=i;
    }
    else if ( max2<fitness[i])
    {
        max2=fitness[i];
        best[1]=i;
    }
}
```

FIGURE 4.7: Selection criteria code

Crossover

The selected chromosomes will be crossed over at each crossover point. In this experiment, we use multiple crossover points. After that, the obtained offspring will be added to the global population, see figure 4.8.

```
for(i=0;i<Nbchrom;i++)
{
    if( max1<fitness[i])
    {
        max2=max1;
        max1=fitness[i];
        best[1]=best[0];
        best[0]=i;
    }
    else if ( max2<fitness[i])
    {
        max2=fitness[i];
        best[1]=i;
    }
}
```

FIGURE 4.8: Genetic Crossover code

Mutation

```
for(int i=0;i<chromsize;i++)
{
        if(Neighbor[i][nbrNodes].f < Neighbor1[i][nbrNodes].f){
            chrom[c][i].x= chrom[c1][i].x ;
            chrom[c][i].y= chrom[c1][i].y ;
        }

}
```

FIGURE 4.9: Genetic Mutation code

Termination Criteria

If the termination criterion is not met, the population that results from the last GA step (i.e., the mutation step) is considered the initial population, and the process keeps repeating till meets this criterion.

4.5 Experimental results and discussion

This section presents the results obtained in our experimental study of the Genetic Algorithm under temperature variations, also a graphical representation (by NetAnim) of nodes deployment in the Area. The used parameters and the obtained results that ensure the near-optimal deployment in a space of 200m*200m under temperature variations are presented as follows.

4.5.1 Parameterization

In order to evaluate the performance of the GA, we propose a deployment area of 200Meters (200m*200m). In this experiment, we use the following assumptions:

TABLE 4.2: Main parameters used

Area	200*200Meters
Number Nodes	200
Number chrom	50

4.5.2 Near-Optimal Deployment under Temperature Effects

This section shows the difference between the initial deployment and the final one after applying the genetic algorithm at different temperature values ($T = -50°C$,$T = 0°C$,$T = +50°C$), which is the near-optimal deployment at the three worst and best variations of the temperature.

Near-Optimal deployment at Tmax = -50

```
-----------------------------------------------Initial Deployment:--------------------------------------------
n_183_86 n_177_115 n_193_135 n_186_92 n_49_21 n_162_27 n_90_59 n_163_126 n_140_26 n_172_136 n_11_168 n_167_29 n_182_130 n_62_123 n_67_135 n_129_2 n_22
_58 n_69_167 n_193_56 n_11_42 n_29_173 n_21_119 n_184_137 n_198_124 n_115_170 n_13_126 n_91_180 n_156_73
Fitness 2

-----------------------------------------------Initial Deployment--------------------------------------------
```

FIGURE 4.10: Initial deployment (Random deployment)

```
-----------------------------------------------Optimal Deployment :--------------------------------------------
best 29
Fitness 352
chrom 29:
n_100_51 n_73_65 n_3_102 n_168_149 n_136_79 n_118_45 n_53_136 n_101_185 n_33_16 n_12_81 n_51_175 n_18_18 n_45_177 n_4_106 n_19_36 n_122_71 n_88_195 n_
88_43 n_49_8 n_145_137 n_87_63 n_182_140 n_199_83 n_78_184 n_52_42 n_65_103 n_128_159 n_121_49

-----------------------------------------------Optimal Deployment --------------------------------------------
```

FIGURE 4.11: Near-Optimal deployment at Temperature $T = -50°C$

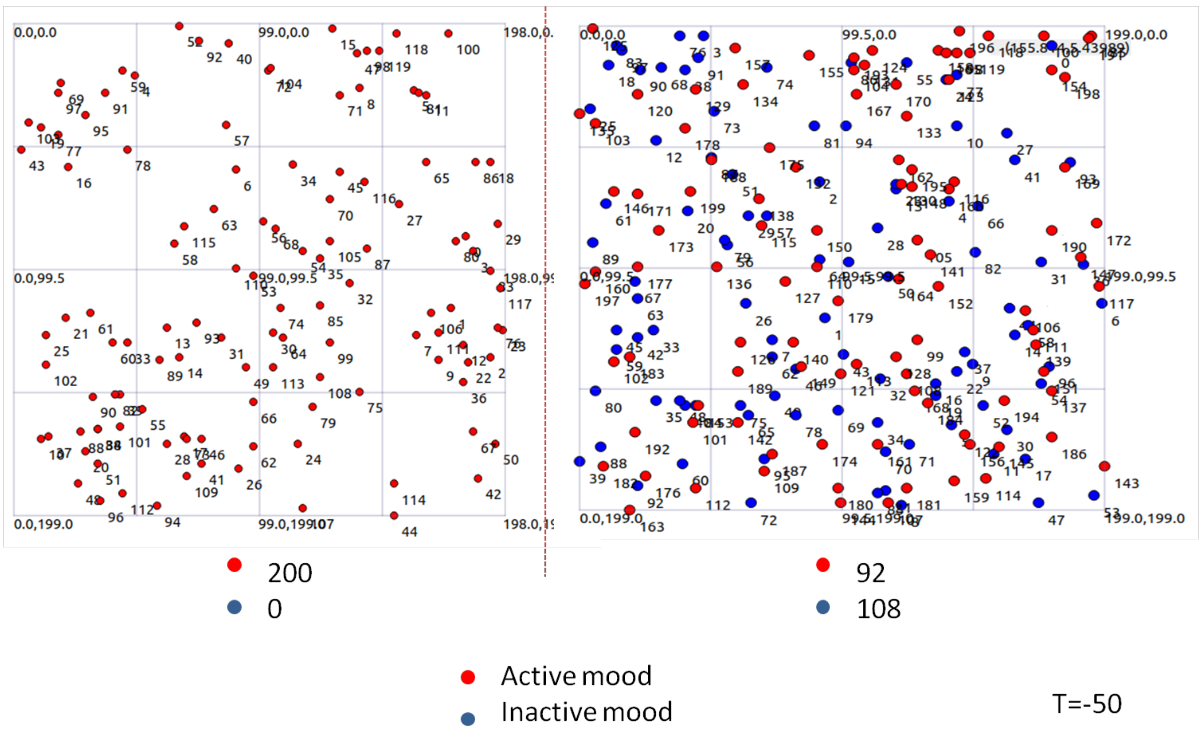

FIGURE 4.12: Initial VS Near-Optimal deployment at $T = -50°C$

Near-Optimal deployment at Tmax = 0

```
-----------------------------------------------Initial Deployment:-------------------------------------------
n_183_86 n_177_115 n_193_135 n_186_92 n_49_21 n_162_27 n_90_59 n_163_126 n_140_26 n_172_136 n_11_168 n_167_29 n_182_130 n_62_123 n_67_135 n_129_2 n_22
_58 n_69_167 n_193_56 n_11_42 n_29_173 n_21_119 n_184_137 n_198_124 n_115_170 n_13_126 n_91_180 n_156_73 n_62_170 n_196_81 n_105_125 n_84_127 n_136_10
5 n_46_129 n_113_57 n_124_95 n_182_145 n_14_167 n_34_164 n_43_150 n_87_8 n_76_178 n_188_184 n_3_51
Fitness 4

-----------------------------------------------Initial Deployment-------------------------------------------
```

FIGURE 4.13: Initial deployment (Random deployment)

```
-----------------------------------------------Optimal Deployment :-------------------------------------------
best 29
Fitness 634
chrom 29:
n_114_176 n_156_37 n_19_141 n_134_45 n_182_27 n_54_195 n_45_86 n_186_142 n_52_152 n_113_11 n_185_138 n_29_4 n_25_84 n_44_94 n_18_157 n_46_40 n_101_46
n_146_196 n_160_12 n_79_180 n_192_87 n_89_107 n_6_100 n_164_159 n_177_106 n_4_15 n_53_38 n_74_8 n_36_87 n_28_7 n_44_74 n_47_98 n_135_196 n_46_172 n_16
0_41 n_152_100 n_9_193 n_8_168 n_94_172 n_127_23 n_30_131 n_190_35 n_169_27 n_124_5

-----------------------------------------------Optimal Deployment -------------------------------------------
```

FIGURE 4.14: Near-Optimal deployment at Temperature $T = 0°C$

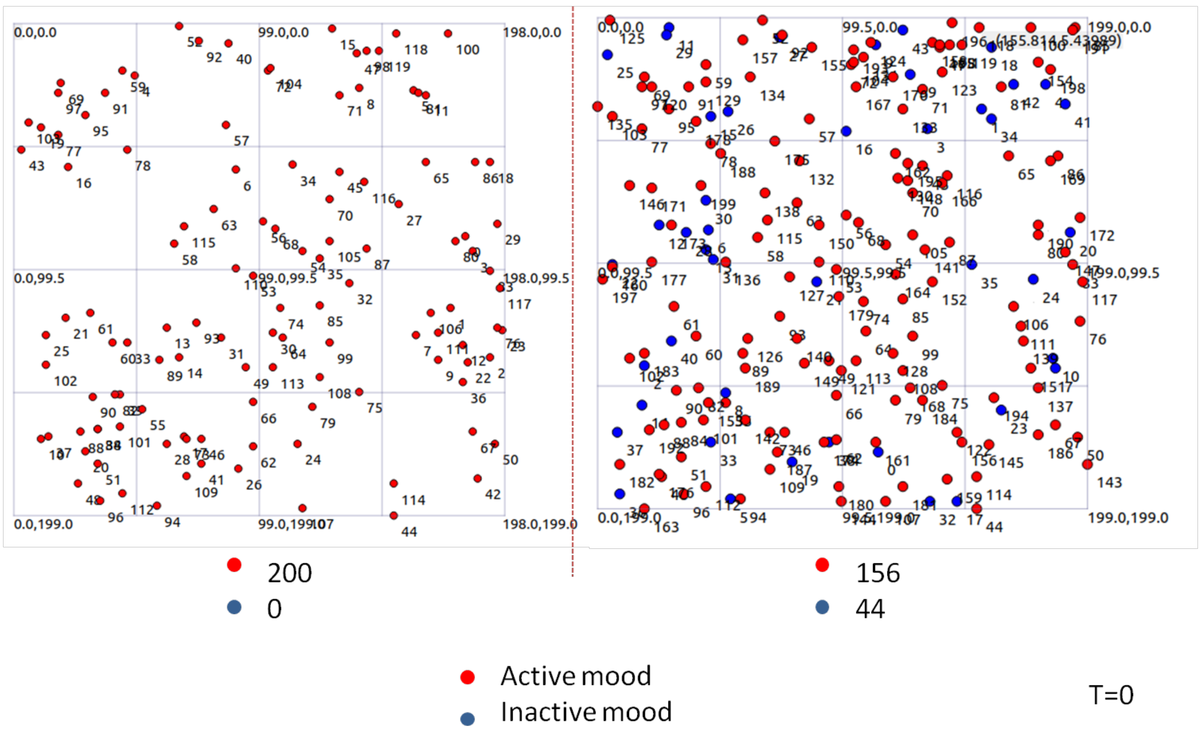

FIGURE 4.15: Initial VS Near-Optimal deployment at $T = 0°C$

Near-Optimal deployment at Tmax = +50

```
-----------------------------------------------Initial Deployment:------------------------------------------
n_183_86 n_177_115 n_193_135 n_186_92 n_49_21 n_162_27 n_90_59 n_163_126 n_140_26 n_172_136 n_11_168 n_167_29 n_182_130 n_62_123 n_67_135 n_129_2 n_22
_58 n_69_167 n_193_56 n_11_42 n_29_173 n_21_119 n_184_137 n_198_124 n_115_170 n_13_126 n_91_180 n_156_73 n_62_170 n_196_81 n_105_125 n_84_127 n_136_10
5 n_46_129 n_113_57 n_124_95 n_182_145 n_14_167 n_34_164 n_43_150 n_87_8 n_76_178 n_188_184 n_3_51 n_154_199 n_132_60 n_76_168 n_139_12 n_26_186 n_94_
139 n_195_170 n_34_178 n_67_1 n_97_102 n_117_92 n_52_156 n_101_80 n_86_41 n_65_89 n_44_19 n_40_129 n_31_117 n_97_171 n_81_75 n_109_127 n_167_56 n_97_1
53 n_186_165 n_106_83 n_19_24 n_128_71 n_132_29 n_103_19 n_70_168 n_108_115 n_140_149 n_196_123 n_18_45 n_46_51 n_121_155 n_179_88 n_164_28 n_41_150 n
_193_100 n_34_164 n_124_114 n_187_56 n_143_91 n_27_165 n_59_136 n_32_151 n_37_28 n_75_7 n_74_121 n_58_195 n_29_37 n_35_193 n_18_28
Fitness 34

-----------------------------------------------Initial Deployment-------------------------------------------
```

FIGURE 4.16: Initial deployment (Random deployment)

```
-----------------------------------------------Optimal Deployment :------------------------------------------
best 29
Fitness 442
chrom 29:
n_179_8 n_93_120 n_91_64 n_47_4 n_140_72 n_141_164 n_198_114 n_73_129 n_122_197 n_149_139 n_143_41 n_157_176 n_29_47 n_120_67 n_165_127 n_102_97 n_135
_147 n_169_178 n_11_16 n_135_152 n_41_76 n_116_191 n_143_142 n_120_65 n_139_22 n_4_34 n_63_114 n_162_44 n_113_83 n_64_78 n_162_166 n_175_97 n_114_145
n_28_125 n_113_163 n_29_154 n_191_98 n_146_134 n_40_18 n_0_179 n_40_156 n_165_55 n_22_128 n_100_135 n_163_116 n_14_125 n_82_141 n_174_196 n_38_154 n_7
4_152 n_117_103 n_58_61 n_153_156 n_195_193 n_175_147 n_124_15 n_56_90 n_71_78 n_170_123 n_14_133 n_39_180 n_10_73 n_73_136 n_22_112 n_91_96 n_64_160
n_151_74 n_21_105 n_31_17 n_98_158 n_116_175 n_125_172 n_65_196 n_51_35 n_71_17 n_61_156 n_38_4 n_143_20 n_82_160 n_55_88 n_6_150 n_89_41 n_150_93 n_1
4_8 n_44_156 n_113_192 n_103_15 n_50_54 n_8_173 n_5_89 n_23_18 n_45_13 n_22_189 n_186_56 n_101_41 n_70_178 n_178_140 n_16_10

-----------------------------------------------Optimal Deployment -------------------------------------------
```

FIGURE 4.17: Near-Optimal deployment at Temperature $T = +50°C$

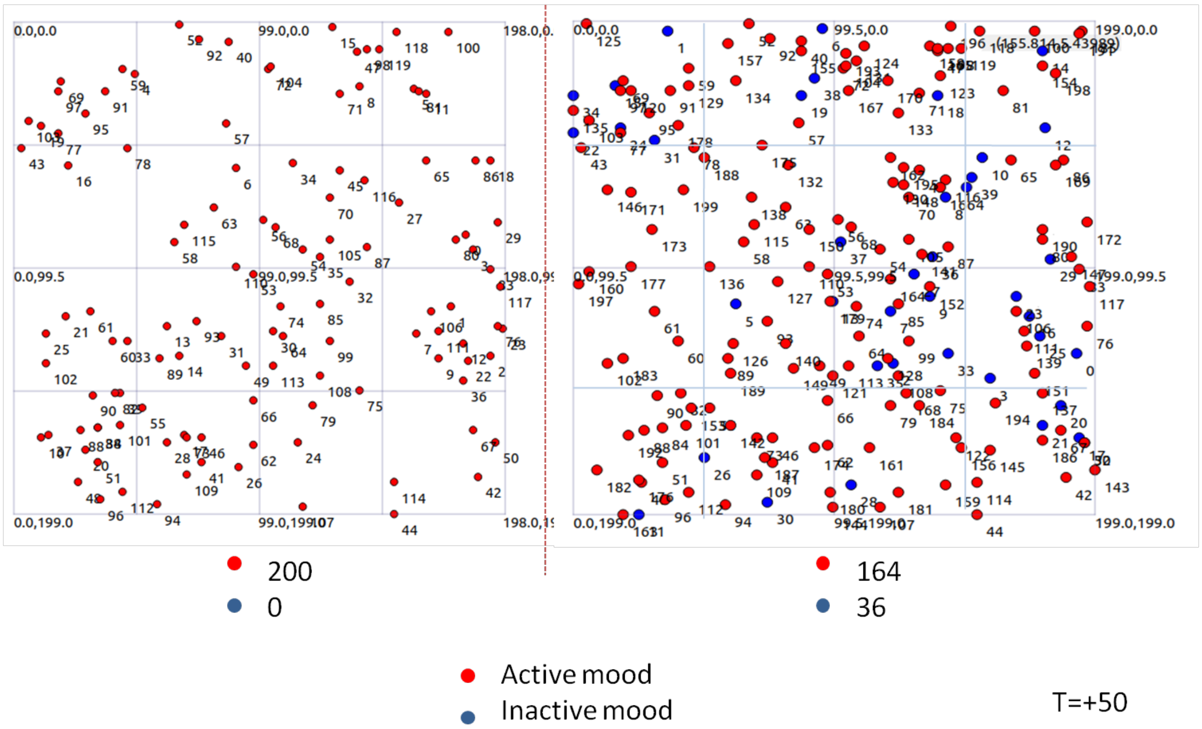

FIGURE 4.18: Initial VS Near-Optimal deployment at $T = +50°C$

4.5.3 Results and discussion

the above figures 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 and 4.16, 4.17, 4.18 represent the obtained results of the difference between WSN deployment at three temperature values ($T = -50°C, T = 0°C, T = +50°C$), as well as the fitness value of the initial and the near-optimal deployment.

In this experiment, we evaluated the results of the Genetic algorithm in which we aim to verify the good working of the algorithm and ensure the improvement of the obtained results between the first and the last generation, as well as to study the impact of the increasing variety of the number of generations on the size of the genetic population.

As showing in these figures 4.10, 4.11, 4.12, 4.13, 4.14, 4.15 and 4.16, 4.17, 4.18 , we can see how the temperature affects the diffusion of sensor nodes in order to guarantee network connectivity and coverage,where when the temperature increases it reduces sensor node communication range R_c, therefore, a more number of active nodes is needed. Inversely, when the temperature decreases it lengthens R_c which requires few active nodes to meet network deployment requirements.

No matter what the temperature degree is, applying the genetic algorithm will make a change in the node's type and location. Wherein in the initial deployment, all nodes are in a state of active mode and after applying the GA we noticed that a large number of nodes changed their location and turned into sleeping mode.

The obtained fitness results in the test phases are very satisfying, where it reaches seven hundred as the best-obtained fitness value refers to the near-optimal WSN deployment.

4.6 Conclusion

This chapter was devoted to the presentation of the programming language and simulation environment. It also details the implementation of the selected method to solve this problem. Moreover, it introduces the obtained results. In order to realize a near-optimal WSN system with evolutionary algorithms, we carefully choose the best combination of tools and platforms. Therefore, the obtained results were promising and reflected in a well-structured system and architecture, such the obtained results are better in terms of coverage, connectivity, and the number of active nodes. While the application of GA in WSN deployment under temperature variations ensures the maximum connectivity with the minimum number of active sensor nodes, which leads to saving more energy and extending the WSN lifetime.

General Conclusion

In this thesis, we apply a genetic algorithm in order to find the near-optimal deployment of WSN with a minimum number of active nodes. We define the near-optimal deployment of WSN as the distribution of sensor nodes that guarantee network connectivity and coverage. However, the optimality of this distribution cannot be guaranteed due to environmental conditions which affect network connectivity, such as temperature variation. Therefore, The near-optimal deployment has to be re-identified each time according to temperature variations, i.e., dynamic deployment. In order to realize this dynamic deployment, sensor nodes have to be mobile nodes. And at each time the set of nodes that do not participate in forming this near-optimal deployment, have to pass in hibernate mode (sleeping mode) to save more energy.

Firstly, we propose a simple coding of GA solutions, as the GA defines the main parameters of the algorithms: mutation and crossover operators, population size, and termination criteria (number of generations). After that, the system decides which sensor nodes should stay active, and which ones should go into a sleeping mood.

The obtained simulation results showed that the genetic algorithm can optimize the random deployment and identify the minimum number of active sensor nodes, also define the best positions of these nodes by enhancing fitness values and maximizing the overall performance from connectivity and coverage. The fitness values obtained during the simulation are very satisfactory, reaching seven hundred in some cases.

In future work, we aspire to apply the proposed GA in the real world in order to make WSN deployment near-optimal and dynamic. We also plan to modify the crossover function to be crossing over multiple points (we have already started working on it, but due to time limitations we could not complete it yet). Moreover, we plan to do a comparison with another optimization method in the way of identify the near-optimal deployment in order to distinguish the best one.

Bibliography

[1] A. Ben, N. Nejah, and A. Kachouri, “Contributions à l’optimisation de la consommation d’énergie dans les réseaux de capteurs sans fil,” Ph.D. dissertation, 01 2013.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] V. Sharma, R. Patel, H. Bhadauria, and D. Prasad, “Deployment schemes in wireless sensor network to achieve blanket coverage in large-scale open area: A review,” *Egyptian Informatics Journal*, vol. 17, no. 1, pp. 45–56, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866515000444

[4] S. Bandyopadhyay and S. Saha, *Some Single- and Multiobjective Optimization Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 17–58. [Online]. Available: https://doi.org/10.1007/978-3-642-32451-2_2

[5] N. Durand, “Algorithmes Génétiques et autres méthodes d’optimisation appliqués à la gestion de trafic aérien,” Habilitation à diriger des recherches, INPT, Nov. 2004. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01293722

[6] V. Kumar, “Overview of strategies and metrics of node deployment in wsn,” 05 2016.

[7] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms for wireless sensor networks: A survey,” *Information Fusion*, vol. 49, pp. 1–25, 2019.

[8] A. Bachir, W. Bechkit, y. Challal, and A. Bouabdallah, “Joint Connectivity-Coverage Temperature-Aware Algorithms for Wireless Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 07, pp. 1923–1936, Jul. 2015. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01006039

[9] M. Numan, F. Subhan, W. Khan, S. Hakak, S. Haider, T. R. Gadekallu, A. Jolfaei, and M. Alazab, “A systematic review on clone node detection in static wireless sensor networks,” 2020.

[10] A. Bendjeddou, "Prolongation de la durée de vie des batteries dans les réseaux de capteurs sans fil (rcsf)," Ph.D. dissertation, Université Badji Mokhtar - Annaba, 2015.

[11] M. Bhattacharjee and S. Gupta, "Determining redundant nodes in a location unaware wireless sensor network," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, pp. 858–862.

[12] A. Förster, "Introduction to wireless sensor networks," 2016.

[13] D. Ngom, "Optimisation de la durée de vie dans les réseaux de capteurs sans fil sous contraintes de couvertureet de connectivité réseau," Theses, Université de Haute Alsace - Mulhouse ; Université Cheikh Anta Diop (Dakar), May 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01531464

[14] H. M. A. Fahmy, *Wireless sensor networks: concepts, applications, experimentation and analysis*. Springer, 2016.

[15] G. Malick, "Etat de l'art sur les wsn(wireless sensor networks)," 2014.

[16] R. Kacimi, "Techniques de conservation d'énergie pour les réseaux de capteurs sans fil," Ph.D. dissertation, 2009.

[17] Y. Challal, H. Bettahar, and A. Bouabdallah, "Les réseaux de capteurs (wsn: Wireless sensor networks)," *Rapport interne, Université de Technologie de Compiègne, France*, 2008.

[18] L. TERRISSA, "Les réseaux du futur réseaux de capteurs sans fil (wsn)," 2015/2016.

[19] A. Fascista, "Toward integrated large-scale environmental monitoring using wsn/uav/crowdsensing: A review of applications, signal processing, and future perspectives," *Sensors*, vol. 22, no. 5, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/5/1824

[20] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[21] "Medecine sans fil." [Online]. Available: http://www.astrosurf.com/luxorion/medecine-sans-fil.html

[22] Y. CHALLAL, "Réseaux de capteurs sans fils," 2008.

[23] Y. Ben Aissa, "Le fonctionnement optimale du réseau de capteurs sans fil sous la variation de la température," 2016.

[24] R. LARBI, "Déploiement optimal des nœuds de capteurs employant le clustering k-means et un algorithme génétique," 2021.

[25] S. Mustapha Reda, "Ceris tnews(le deploiement des reseaux de capteurs sans fil)," 2014.

[26] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Comput. Surv.*, vol. 43, no. 4, oct 2011. [Online]. Available: https://doi.org/10.1145/1978802.1978811

[27] I. Khoufi, P. Minet, A. Laouiti, and S. Mahfoudh, "Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges," *International journal of autonomous and adaptive communications systems*, vol. 10, no. 4, pp. 341–390, 2017.

[28] D. He, J. Portilla, and T. Riesgo, "A 3d multi-objective optimization planning algorithm for wireless sensor networks," in *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2013, pp. 5428–5433.

[29] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[30] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 2. IEEE, 2003, pp. 1293–1303.

[31] O. Abdelraouf, B. Miloud, L. Noureddine, M. Boushra, and B. Fatima Zohra, "Réseaux de capteurs sans fils." [Online]. Available: https://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_14.html

[32] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on evolutionary computation*, vol. 7, no. 2, pp. 117–132, 2003.

[33] L. SAADI, "Optimisation multiobjectifs par programmation génétique," Juillet 2007. [Online]. Available: http://eprints.univ-batna2.dz/226/

[34] S. Bourazza, "Variantes d'algorithmes génétiques appliquéees aux problèmes d'ordonnancement," Theses, Université du Havre, Nov. 2006, 24/01/07. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00126292

[35] H. L. Sidi Mohamed Douiri, Souad Elbernoussi, "Cours des méthodes de résolution exactes heuristiques et métaheuristiques."

[36] K. FADIA, "Optimisation multi-objectif du déploiement des réseaux sans fil dans les serres intelligentes."

[37] C.-L. Tseng and W.-H. Chang, "Coverage hole repairment for mobile wireless sensor networks with simultaneous multiple node deaths," in *2021 International Automatic Control Conference (CACS)*. IEEE, 2021, pp. 1–5.

[38] M. Choudhary and N. Goyal, "Dynamic topology control algorithm for node deployment in mobile underwater wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 15, p. e6942, 2022.

[39] A. Ndam Njoya, W. Abdou, A. Dipanda, and E. Tonye, "Approche Évolutionnaire de déploiement des capteurs sans fil pour la couverture des cibles," 10 2015.

[40] B. Foubert and N. Mitton, "Sélection d'interface de communication dans les réseaux de capteurs multi-technologies," in *CoRes 2020 - 5èmes Rencontres Francophones sur la Conception de Protocoles, l'Evaluation de Performance et l'Expérimentation des Réseaux de Communication*, Lyon, France, Sep. 2020. [Online]. Available: https://hal.inria.fr/hal-02570838

[41] A. Al-Hourani, "How can i compute the energy consumed due to mobility in mobile wireless sensor networks?" 04 2015.

[42] "C++ tutorial." [Online]. Available: https://www.w3schools.com/CPP/cpp_intro.asp

[43] "Qt creator - a cross-platform ide for software development." [Online]. Available: https://www.qt.io/product/development-tools

[44] F. C. BOUMGHAR, W. BACHA, M. CHAIANI, and B. Dalila, "Simulateurs de réseaux ns-3 et ns-2," 2014.

[45] "Netanim 3.108." [Online]. Available: https://www.nsnam.org/wiki/NetAnim