



Modeling, Scheduling and Optimization of Wireless Sensor Networks lifetime

Yousif Elhadi Elsideeg Ahmed

► To cite this version:

Yousif Elhadi Elsideeg Ahmed. Modeling, Scheduling and Optimization of Wireless Sensor Networks lifetime. Performance [cs.PF]. Université de Lorraine, 2016. English. NNT : 2016LORR0315 . tel-01543080

HAL Id: tel-01543080

<https://theses.hal.science/tel-01543080>

Submitted on 20 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



Modeling, Scheduling and Optimization of Wireless Sensor Networks lifetime

by

Yousif Elhadi Elsideeg Ahmed

Submitted to the IAEM Lorraine Doctoral School in fulfillment of the requirements for the degree of

DOCTORATE

in

Automation, Signal and Image Processing, Computer Engineering

Dissertation Committee

Imed KACEM	Prof. Université de Lorraine	President
Feng CHU	Prof. U. d'Evry Val d'Essonne	Rapporteur
Mhand HIFI	Prof. U. de Picardie Jules Verne	Rapporteur
Marie-Ange MANIER	MDC HDR U. de Technologie de Belfort-Montbéliard	Rapporteur
Zineb SIMEU-ABAZI	MDC HDR Grenoble Université	Examiner
Magdi B. M. AMIEN	Associate prof. U. of Gezira	Examiner
Kondo H. ADJALLAH	Prof. Université de Lorraine	Supervisor
Sharief F. BABIKIR	Prof. U. of Gezira	Co-supervisor

December 2016

Modeling, Scheduling and Optimization of Wireless Sensor Networks lifetime

Yousif Elhadi Elsideeg Ahmed

Abstract

Wireless sensor networks (WSNs), as a collection of sensing nodes with limited processing, limited energy reserve and radio communication capabilities, are widely implemented in many areas of applications such as industry, environment, health-care, etc. Regarding this large range of applications, many research issues are introduced including the applications, performance, reliability, lifetime, etc. The WSNs lifetime considered in this work is the period of time through which the WSN is perfectly completing its function. This lifetime is affected by many factors including the amount of energy available, failure probability and components degradation. The amount of energy available become the most important factor in case of non renewable components applications. Different algorithms, strategies and optimization techniques were developed and implemented for this purpose based on the possibility of activating a subset of sensors that satisfied the monitoring constraint, while keeping the others in sleep mode to be implemented later. This is an NP complete maximization problem that can be solved using disjoint set covers (DSCs). But the solution obtained using DSCs does not extend always significantly the WSNs lifetime. So, the present work aims to search for a better solution using non-disjoint set covers (NDSCs). This approach gives the opportunity for a sensor to be implemented in one or more subset covers. For that purpose, we studied a binary representation based model to maximize the number of NDSCs. Also, we developed a genetic algorithm based heuristic based on this model to find out the maximum number of NDSCs in a reasonable time. Thus, for a set of m sensors used to monitor a set of n targets or a field, this heuristic allows to construct a maximum number q of NDSCs. Additional effort is required to find the best scheduling for implementing the NDSCs so as to maximize the lifetime of the sensors involved in the WSNs, considering their limited available energy. This problem is formulated using integer linear programming (ILP) mathematical model. The objective function of this problem is the sum of all monitoring seasons on which all q NDSCs scheduled, and the constraint is the energy consumption in

all sensors included in all NDSCs. Solving this problem using ILP in a period of time depends on the complexity of the model and the instances used. To find the solution in reasonable time, we have developed a NDSCs based genetic algorithm (NDSC-GA). The candidate solutions are represented in chromosomes composed of a number of genes equal to the number q of NDSCs, and each gene is the number of monitoring seasons on which a NDSC is scheduled. We have then developed a GA that combines the four crossover operators and four mutation operators. The GA based methods are coded in C programming language to obtain a satisfying solution and the Cplex software was used to obtain the corresponding exact solution. Comparing the optimal solution obtained using the ILP on small instances, to the solutions obtained using our GA based method explained that our methods can find a solution near the optimal in reasonable time. Then, comparing the solution obtained using our NDSCs GA based methods, to the DSCs GA based method in the literature, we showed that the NDSCs GA can prolong the WSNs lifetime better than DSCs GA for the same instances. Our approach combines together the scheduling principles and the optimization techniques to maximizing the WSNs lifetime.

Modélisation, Ordonnancement et Optimisation de la Durée de Vie des Réseaux de Capteurs Sans Fil

Resumé

Les réseaux de capteurs sans fil (RCSFs), sont composés d'un ensemble de nœuds avec des capteurs, transmetteur/récepteur, d'un système de traitement et d'une réserve d'énergie. Au regard d'applications, de travaux de recherche sont développés sur l'utilisation de ce réseau leur performance, fiabilité ou durée de vie. La durée de vie RCSFs correspond à la période à travers laquelle le RCSF fonctionne parfaitement. Cette durée de vie est très affectée par de nombreux facteurs comme la quantité d'énergie disponible, la probabilité de défaillance et les dégradations des composants. L'énergie disponible devient le facteur prépondérant dans les cas d'applications avec des composants difficilement rechargeables ou non renouvelables. Différents algorithmes, stratégies et techniques d'optimisation ont été élaborées et mises en œuvre à cet effet sur la possibilité d'activer un sous-ensemble de capteurs qui satisfont à la contrainte de surveillance et de garder les autres capteurs en mode veille pour pouvoir être mis en œuvre ultérieurement. Ainsi, c'est un problème de type NP complet de maximisation qui peut être résolu en considérant des Ensembles Disjoints de capteurs de Couverture (EDC). Mais la solution obtenue à l'aide des EDCs ne conduit pas toujours à une extension significative de la durée de vie des RCSFs. Le présent travail vise à rechercher une meilleure solution basée sur des capteurs regroupés dans des ensembles non-disjointes de couverture (ECND). Cette approche permet à un capteur de participer à une ou plusieurs ensembles de capteurs de couvertures. Nous avons alors étudié un modèle de représentation binaire des ECNDs pour déterminer un ordonnancement optimum permettant de maximiser la vie d'un RCSF. De plus, nous avons développé une heuristique basée sur un algorithme génétique, pour trouver une solution proche de l'optimal dans un délai raisonnable. Ainsi, pour un ensemble de m capteurs utilisés pour surveiller un ensemble de n cibles, cette heuristique permet construire un nombre maximum q d'ensembles ECNDs. Des efforts supplémentaires sont donc nécessaires pour trouver le meilleur ordonnancement pour la mise en œuvre des ECNDs, qui maximise la durée de vie globale

du RCSF, compte tenu de l'énergie initialement disponible dans chaque capteur. Ce problème est formulé à l'aide d'un modèle mathématique de programmation linéaire en nombres entiers (PLE). La fonction objective de ce problème est la somme de toutes les périodes de surveillance pour les q ECNDs programmés, et la contrainte est la consommation d'énergie de tous les capteurs constituant les ECNDs. La possibilité de trouver la solution à ce problème par PLE dans une période de temps donnée dépend de la complexité du modèle et des instances utilisées. Pour trouver la solution dans un délai raisonnable, nous avons développé un algorithme génétique (AG) basé sur les ECNDs. Les solutions potentielles sont représentées dans des chromosomes composés d'un certain nombre de gènes correspondant aux ECNDs, et chaque gène est caractérisé par la période de surveillance d'un ECND. Nous avons ensuite développé un AG qui combine quatre opérateurs de croisement et quatre opérateurs de mutation. La méthode basée cet AG a été codée dans le langage de programmation C pour obtenir une solution satisfaisante et le logiciel Cplex a été utilisé de déterminer la solution exacte correspondant. Une comparaison des solutions obtenues sur de petites instances en utilisant la PLE par rapport aux solutions obtenues par notre AG montre que la méthode basée sur les AG peut trouver une solution proche de l'optimale dans un délai raisonnable. Ensuite, en comparant les solutions en utilisant l'AG ECNDs à l'AG EDCs de la littérature, nous montrons que l'AG avec ECND peut prolonger la durée de vie des RCSFs plus que les AG avec EDCs pour les mêmes instances. Notre approche combine ainsi les principes d'ordonnancement et les techniques d'optimisation pour maximiser la durée de vie des RCSFs.

Acknowledgements

It is my great pleasure to express my sincere and deep thanks and gratitude to my supervisors Prof. Kondo H. ADJALLAH and Prof. Sharief F. BABIKIR for the continuous help, support, patience and advice in all the time of my Ph.D research and writing of this thesis, Thank you for knowledge, efforts, time, friendship and easy to communicate.

I would like to express my sincere and deep thanks to Prof. Imed KACEM for his administrative support to establish this work and his fruitful scientific advice.

Equally, my sincere and deep thanks goes to my thesis jury members for their efforts, advices and comments.

My sincere and deep thanks goes to the vice chancellor of the university of Gezira for his administrative efforts and support.

My sincere and deep thanks goes to the Embassy of France in Sudan administrative efforts, help and support.

Equally, I would like to express my sincere and deep thanks to my colleagues in LCOMS, ENIM and FET, Sisters, brothers and friends for their continuous support.

Last but not the least, I would like to thank and appreciate my family's stand beside me: my parents Elhadi and Aisha for my being here and supporting me kindly throughout my life. My wife Budria for unlimited support.

I would like to express my sincere and deep wishes to my small and perfume roses: Mohamed, Ibraheem, Ibaa and Hadi with safe, success and happy journey through the life.

List of Publications

- Yousif Elhadi E. Ahmed, Kondo H. Adjallah, Romuald Stock, Imed Kacem and Sharief F. Babikir, Exact and Heuristic Methods for Maximizing Lifespan of Randomly Deployed Wireless Sensor Networks for Long-life Infrastructures Monitoring, Summited to: CAIE journal, 2016.
- Yousif Elhadi E. Ahmed, Kondo H. Adjallah, Romuald Stock and Sharief F. Babikir, Wireless Sensor Network Lifespan Optimization with Simple, Rotated, Order and Modified Partially Matched Crossover Genetic Algorithms, The 14th IFAC conference on programmable devices and embedded systems (PDES2016), Brno, Czech Republic, 2016.
- Yousif Elhadi E. Ahmed, Kondo H. Adjallah and Sharief F. Babikir, Non Disjoint Set Covers Approach for Wireless Sensor Networks Lifetime Optimization, The 3rd Symposium on Wireless Systems within the IEEE the IDAACS 2016, Offenburg, Germany, 2016.
- Yousif Elhadi E. Ahmed, Kondo H. Adjallah, Imed Kacem and Sharief F. Babikir, Integer Linear Programming Based Scheduling Method For Wireless Sensors Network Lifespan Optimization, CIE45, Metz, France, 2015.
- Yousif Elhadi E. Ahmed, Kondo H. Adjallah, Imed Kacem and Sharief F. Babikir, Genetic Algorithm based Scheduling Method for Lifespan Extension of a Wireless Sensors Network, IDAACS Warsaw, Poland, 2015.

Contents

Abstract	i
Resumé	iii
Acknowledgements	v
List of Publications	vi
Contents	vii
List of Figures	x
List of Tables	xii
General Introduction	1
1 State of the Art on WSNs Scheduling and Lifetime Optimization	5
1.1 Overview	5
1.2 Introduction to Wireless Sensor Networks	6
1.2.1 WSNs applications	9
1.2.2 Wireless Sensor Networks research challenges	14
1.2.3 Wireless Sensor Networks description and modeling	16
1.2.4 WSNs lifetime optimization through energy consumption	17
1.2.5 WSNs reliability, failure and self-adaptivity	18
1.3 Scheduling problem	19
1.3.1 Modeling the scheduling problems in WSNs	20
1.3.2 Scheduling environments, constraints, and objectives	21
1.3.3 Complexity of the scheduling and optimization problems	23
1.3.4 Scheduling classes	24
1.4 Optimization approaches	25
1.4.1 Combinatorial optimization	27
1.4.2 Optimization methods	28
1.4.2.1 Exact methods	28
Linear Programming	28
Branch and Bound	30
1.4.2.2 Heuristics and meta-heuristics	30

	Genetic Algorithm	31
	Simulated Annealing:	35
1.5	Conclusion	37
2	Problem Statement as Disjoint Set Covers Maximization	39
2.1	Overview	39
2.2	Problem statements and description	40
2.3	Related works	44
	2.3.1 Disjoint Set Cover based methods	45
	2.3.2 Exact methods	50
	2.3.3 Heuristics and Meta-heuristics based methods	53
2.4	Conclusion	56
3	Methods for Solving the Problem as Non-Disjoint Set Covers	59
3.1	Introduction	59
3.2	Non-Disjoint Set Covers finding strategies	62
	3.2.1 The NDSC contribution to the DSC	62
	3.2.2 The binary representation based method	64
	3.2.3 Genetic Algorithm based method	66
3.3	Scheduling and optimization strategies	70
	3.3.1 The mathematical model	71
	3.3.2 Integer Linear Programming based method	75
	3.3.3 Genetic Algorithm based method	78
	3.3.3.1 Crossover operators	81
	3.3.3.2 Mutation operators	83
	3.3.3.3 GA configurations	87
3.4	An existing DSC based method	88
3.5	Conclusion	89
4	Evaluation of the Methods through Numerical Simulation	91
4.1	Overview	91
4.2	The binary representation method for finding the NDSCs	92
4.3	The GA based method for finding the NDSCs	94
4.4	ILP based method simulation and results	96
4.5	GA based method simulation and results	102
4.6	The existing method simulation and results	111
4.7	Results analysis and evaluation	112
	4.7.1 Evaluating the GA based strategies results	113
	4.7.2 Evaluating the GA to the Optimal	115
	4.7.3 Evaluating the NDSCGA to the DSCGA	117
4.8	Conclusion	119
	General Conclusion	121
	Conclusion Générale	125

Bibliography

128

List of Figures

1.1	Wireless Sensor Node [1] [2].	6
1.2	Sensors connected through internet via base stations.	7
1.3	WSNs Protocol Stack [3].	7
1.4	ZigBee Protocol Stack [3].	8
1.5	An ad-hoc networks [4].	9
1.6	Application and a service interface [5].	10
1.7	IoT future and applications [6].	13
1.8	IoT research challenges [7].	13
1.9	WSNs deployment.	17
1.10	The self-adaptivity common stages	19
1.11	Sensor to targets coverage scheduling	20
1.12	Covers to targets coverage scheduling	21
1.13	Complexity classes.	24
1.14	Scheduling classes.	25
1.15	Maximization and minimization of $f(v)$	26
1.16	Evolutionary algorithm applications.	27
1.17	Graphical method for simple LP - unique solution.	29
1.18	Graphical method for simple LP - parallel function.	30
1.19	Genes and chromosomes	32
1.20	A simple evolution cycle.	32
1.21	Single point crossover and two point crossover	34
1.22	The partially matched crossover	34
1.23	The simple genetic algorithm	36
2.1	Sensors and targets relation.	43
2.2	Sensors selection in covers.	46
2.3	Sensors and possible covers.	48
2.4	Covers scheduling.	49
2.5	Two sensing ranges sensors and targets relations.	52
2.6	Sensors activation/deactivation.	53
2.7	Sensors node status.	54
2.8	Directional sensors.	56
3.1	a schduling formulation of WSNs lifetime optimization	61
3.2	The method owerview	63
3.3	Covers scheduling.	73

3.4	Genetic algorithm chromosome creation.	79
3.5	Genetic algorithm crossover strategies.	80
3.6	The Genetic algorithm Combinations	81
3.7	The order crossover	83
3.8	Rotated Crossover	84
3.9	Genetic algorithm main stages.	87
3.10	GA Configurations	88
3.11	DSC encoding in GA.	88
4.1	The number of cover for small and greater population size	95
4.2	GA11 vs GA12.	105
4.3	Deterministic vs Randomize GA11.	106
4.4	The WSNs lifetime using SX, PMX, RX and OX crossover with p=10.	107
4.5	WSNs lifetime using SX, PMX, RX and OX crossover with p=100.	107
4.6	The lifetime using one and two points mutations.	108
4.7	The lifetime using deterministic and randomized mutations.	108
4.8	The energy consumption via all sensors.	109
4.9	The 21 covers utilization via different number of generations.	109
4.10	Energy consumption by all sensors using SX, PMX, RX and OX crossovers.	109
4.11	The 21 covers utilization via different number of generations using SX, PMX, RX and OX crossovers.	110
4.12	The lifetime for WSN with not identical sensors.	110
4.13	The DSCs for different numbers of sensors.	111
4.14	The DSCs for different numbers of iterations.	112
4.15	Crossover one-point randomized with different number of iterations.	113
4.16	Crossover two-point randomized with different number of iterations.. . . .	114
4.17	Crossover one-point deterministic with different number of iterations.. . . .	114
4.18	Crossover two-point deterministic with different number of iterations.. . . .	115
4.19	GA with NDSC via GA with DSC.	118

List of Tables

2.1	Problem notations	40
2.2	The sensor deployment and the targets positions	42
3.1	Initial population	86
4.1	The relation between the population size and number of NDSCs . .	94
4.2	The relation between the population size and number of NDSCs (ng=20)	95
4.3	The WSN expandability	95
4.4	GAMSC and GANDSC in ms	96
4.5	SILSOM and ILSOM	100
4.6	ILSOM for 10 sensors	100
4.7	ILSOM for 20 sensors	102
4.8	ILP model computation time samples	102
4.9	Different GA mutation strategies	104
4.10	GA and ILP based mettods comparison	116
4.11	MSCGA and NDSCGA	118
4.12	DSCGA and NDSCGA	119

General Introduction

A wireless sensor network (WSN) is a collection of a vast number of small, low-cost, low-power and multi-functional sensor nodes deployed over a region or embedded in a target to be monitored or tracked. Each sensor node consists of a processing capability, a memory unit, an RF transceiver, an electrical battery as power source, and accommodate various sensors and actuators [8]. These nodes self-organize in a cooperative network [9] to communicate and transmit the sensor measurements to the end user. The lifetime of each sensor node depends on the energy stored in the electrical battery. The sensor is considered to be dead once the battery is exhausted. Most of the applications of WSNs are intended to monitor a region or a set of targets. In some applications, targets may be located in a dangerous or remote area where installing sensors in specific positions can be a difficult or impossible task. In this case, sensors could not be accessed when installation is completed. For such difficult or impossible to deploy WSNs applications, sensors are randomly deployed in large numbers by flying an aircraft over the region to be monitored to ensure that the area or targets of interest could be covered. The network lifetime is defined as the time elapsed until any active sensor set fails to satisfy the required coverage [10]. Possible primary states of a sensor in WSN can be either active or in sleep, where active state consists of three possible states: transmitting signal, receiving signal and sleeping (or idle waiting for send/receive). To extend the lifetime of a sensors network, minimal subsets of sensors can actively cover the targets, while the other sensors can sleep. Then, the problem is to determine how long to use a given subset and which subset to use next as a scheduling approach [11]. A significant number of researchers addressed the issue of efficient

energy management in wireless sensor networks considering the new constraints about sensing coverage introduced to satisfy the distributed nodes sensing requirements [12]. Powerful and modern optimal scheduling methods have emerged for solving complex engineering optimization problems in the recent years regarding the various evolutionary computation methods addressed. These methods include mathematical programming techniques, genetic algorithms, simulated annealing, ant colony optimization, neural network-based optimization, fuzzy optimization, etc. The optimization problem can be solved by using decision data, the objective function to be optimized and the constraints to be met [13]. In GAs, the term chromosome is typically referred to a candidate solution for an optimization problem, often encoded as a string of numbers, characters or bits. The genes are either single digit or short blocks of adjacent bits or characters that encode a particular element of the candidate solution [14].

Considering the WSNs application in which the sensors are not rechargeable, the battery lifespan is the available period of sensor node utilization. Therefore, the optimal lifetime for such WSNs is exactly the optimal utilization time of this limited resources. For a set of sensors used to monitor a set of targets or region, subsets of sensors that satisfy the required monitoring should be found so as to be scheduled and implemented to prolong the network lifetime. The current work is an investigation for modeling and optimizing the life of such like set of sensors used for monitoring a set of targets or some fields. It aims to formulate the mathematical model of this problem through which the optimal energy utilization could be planned and the optimal lifetime could be obtained. This work tried to implement the mathematical programming and the evolutionary algorithm to build an efficient method for WSNs lifetime optimization, considering limited initial energy for the involved sensor nodes. An integer linear programming (ILP) model is developed, and the GA is used in this work to solve the problem of randomly deployed wireless sensors network lifetime optimization formulated as scheduling problem.

The rest of this thesis is planned as follows:

- In chapter 2, the literature review of this problem is covered considering the

WSNs, the scheduling, and the optimization theory. The WSN is described regarding the architecture, protocols, applications and research challenges. Then, the models, classes, environment, objectives, and complexity of the scheduling problems are presented. Finally, the optimization theory is introduced considering both exact and heuristic methods that are aimed to be implemented in this work. The most common exact methods of the linear programming (LP) and branch and bound (B&B) are briefly presented in addition to the most common heuristic and meta-heuristic methods such as greedy algorithm and genetic algorithms (GA).

- In chapter 3, we stated the problem of WSNs lifetime regarding the environmental constraints and the objective function to be maximized. Then, the most popular methods used to solve this problem in the literature are described. It is clear that the disjoint set cover maximization (DSC) problem is widely used to solve the WSNs lifetime optimization problem. Different exact and heuristic methods are used for WSNs lifetime optimization formulated as DSCs maximization problem. The WSNs lifetime problem is formulated in many different ways of mathematical modeling and different exact, and heuristic methods are used to solve it. One should notice that this problem is NP-hard.
- Chapter 4 details the method we developed using the non-disjoint set covers (NDSCs) approach. For solving this problem, we split it into two sub-problems: 1) find the maximum number of NDSCs and 2) find the optimal scheduling that maximizes the WSNs lifetime. To find the optimal number of NDSCs we used a simple heuristic developed for this propose then worked out a GA based one. The second sub-problem started with the mathematical modeling of the problem then we developed an integer linear programming algorithm to find an optimal solution. Finally, we suggested and developed a GA for searching a near optimal solution in reasonable time. The GA based method coding, initialization, fitness, crossover, mutation, and selection functions are detailed. Several possible configurations of the GA are possible based on four mutation strategies and for crossover operators.

Chapter 1

State of the Art on WSNs Scheduling and Lifetime Optimization

1.1 Overview

This chapter aims to present an integrated vision of the problems incorporating the scheduling and the combinatorial optimization in the area of wireless sensor networks. The wireless sensor networks (WSNs), the scheduling, the optimization and the recent research work in this scope should be explained. A general description of WSNs then, applications and the recent related research issues are considered. The scheduling area is briefly presented considering the models, the complexity, the algorithms and the operation. This chapter focuses on the combinatorial optimization methods including the exact methods, the heuristic and meta-heuristics, such as the integer linear programming and the genetic algorithms implemented in this work. To sum up, this chapter helps to understand the problem stated, with its related works in the next chapter.

1.2 Introduction to Wireless Sensor Networks

The sensing nodes are low-cost devices embodying a unit of digital signal processors (DSP), with low-power radio frequency (RF) communication capability, and energy stored in a small battery [1] [2] (see figure 1.1). The nodes have the ability to collect and communicate data to each other, or to a base station. Thus, each node can send data through the network for various utilizations such as monitoring, or decision support.

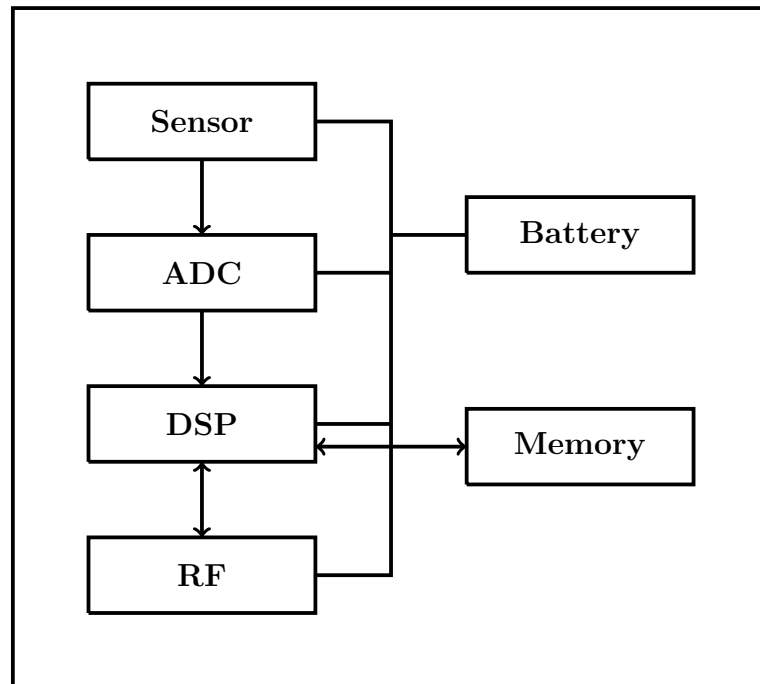


FIGURE 1.1: Wireless Sensor Node [1] [2].

Sensor nodes communicate not only with each other but also with a base station (BS), using their wireless radios, allowing them to disseminate their data to remote systems of data processing, visualization, analysis, and storage. Figure 1.2 illustrates two sensor networks assigned to monitoring two distinct areas and connected through the intranet, using their base stations [15] [3]. Authors in [16] have described different interconnection architectures.

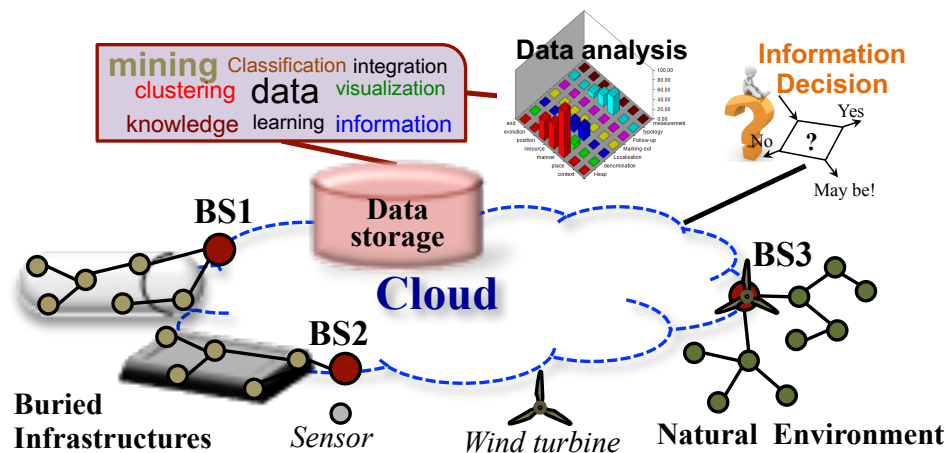


FIGURE 1.2: Sensors connected through internet via base stations.

In addition to the communication protocol layers (application, transport, network, data-link, and physical), the management (mobility, quality of service, security and power management) challenges should be considered in the WSNs [17]. The extended version of this standard was presented by Wang and Balasingham in [3] as in figure 1.3.

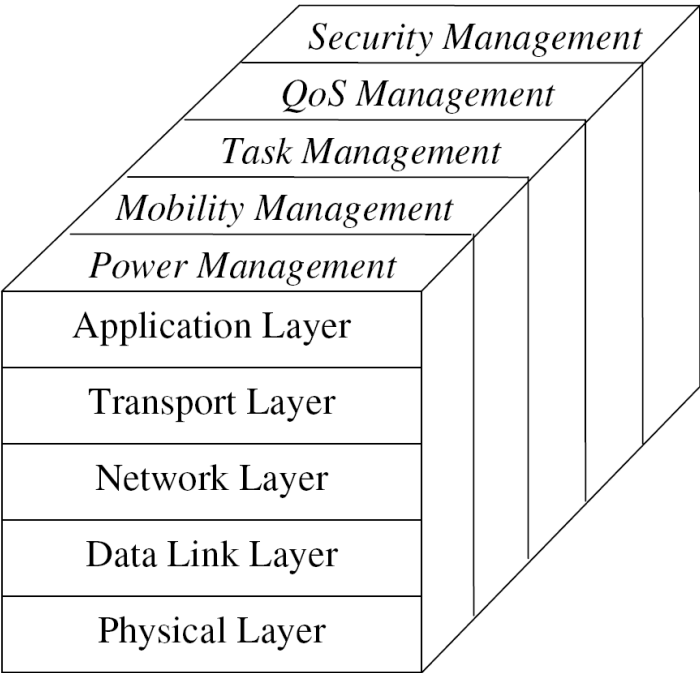


FIGURE 1.3: WSNs Protocol Stack [3].

The IEEE 802.15.4 standard specifies the physical layer and medium access control (MAC) layer characteristics of low power and low data rate radio communications used in WSNs such as the ZigBee standard. The features based on the IEEE 802.15.4 standard include data rates of 250 kbps, 40 kbps, and 20 kbps, two addressing modes (16-bit short and 64-bit IEEE for addressing), Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) is used for channel access. Fully handshake protocol for transfer reliability, power management for low energy consumption, 16 channels in the 2.4GHz ISM band, 10 channels in the 915MHz for the industrial, scientific and medical radio (ISM) band, and one channel in the 868MHz band [3].

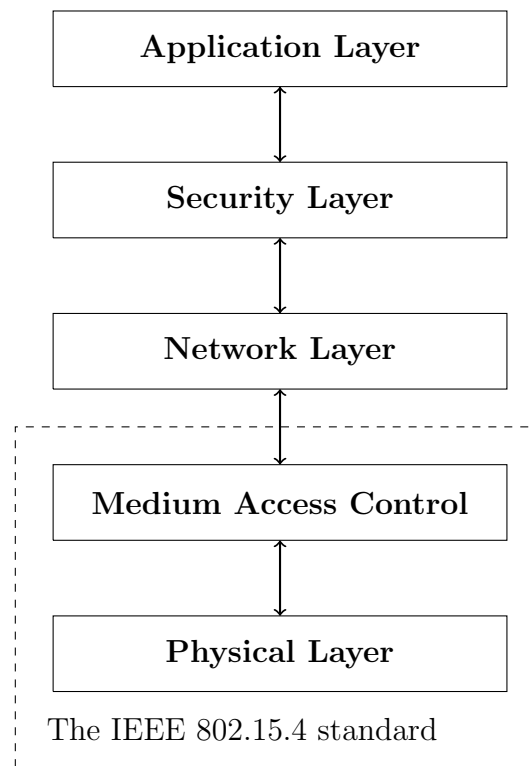


FIGURE 1.4: ZigBee Protocol Stack [3].

Based on the IEEE 802.15.4 standard, ZigBee is a low-cost, low-complexity and low power technology. The characteristics of this technology allow to developing full wireless mesh networks, involving up to 65,000 nodes in the wide range industry networks. It has network layer, security layers and application layers in addition to the IEEE 802.15.4 standard (see figure 1.4). It has a global (2.4 GHz) and regional (915MHz Americans) and (868 MHz Europe) operation bands, and it has

various transmission options and security key generation mechanism based on the Advanced Encryption Standard (AES-128) security scheme [18] [19].

Different components and algorithms could be implemented in each of these layers, thus giving the opportunity to WSNs to be widely applied as an efficient link between the digital virtual world and the physical world. Efficient multi-channel media access control could apply to establish the nodes connectivity [4].

The ad-hoc networks [4], depicted in Figure 1.5, offered a flexible access medium, with greater energy efficiency, security, etc., while the capability Using custom routing control protocols. This kind of network is selected for a significant amount of research investigations and many applications [20] [21].

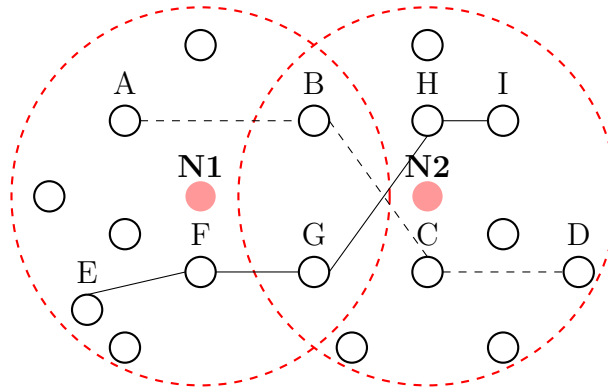


FIGURE 1.5: An ad-hoc networks [4].

The following subsections describe some of the WSNs applications, the research challenges, and the WSNs modeling. Also, the problem of lifetime optimization is introduced through the main factors affect the WSNs lifetime such as the energy, components degradation and failure.

1.2.1 WSNs applications

WSNs applications are increasingly penetrating into a broad range of the daily life and systems with a large variation in characteristics and requirements. Their operation relies on exchanges of data and information through different layers using protocol stacks with specific constraints and needs. So, to meet the increasing

needs of users and applications, [22] and solve the tough operating problems, close collaborations will be required between software developers, researchers, and hardware designers.

From service provider vision, a WSNs aims to provide the quality of service (QoS) required for the application layer, considering the physical and data link layer constraints. In a more abstracted vision, when the hardware constraints related to the application, the physical and data link layer, and the services are encapsulated, a service interface abstraction level is used for the interaction between the requests and the WSN, and the ad-hoc architecture may be presented as in figure 1.6 [5].

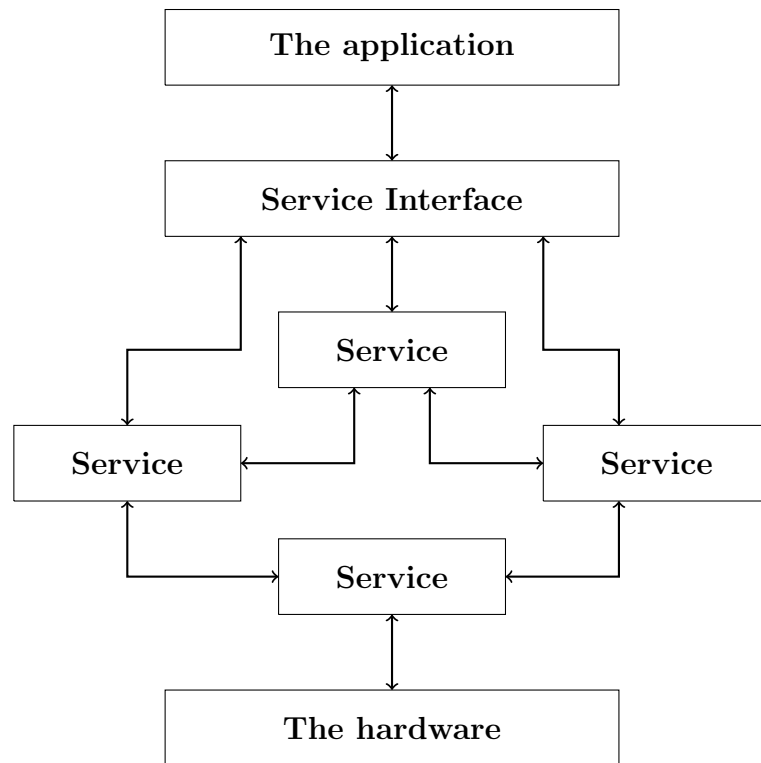


FIGURE 1.6: Application and a service interface [5].

The WSNs applications include environmental monitoring, industrial infrastructures, civil infrastructures, logistics, military, positioning and tracking, transportation, medical applications, cyber-physical systems and the internet of things [5] [22].

- **Environmental applications**

The WSNs are intensively implemented in the environmental monitoring applications such as weather, radiation and air pollution monitoring systems [23]. The WSNs architecture and nodes should have the capability to communicate with each other, collect the environmental data from a region of interest and transmit the data via the gateway to the monitoring center in addition to the limited processing capability. The processing can minimize the invalid data transmission to reduce the overall energy consumption during data transmission. The WSNs can monitor several environmental parameters according to the application such as underground water level, pressure, temperature, wind direction and speed to provide various services for end users. The architecture of such systems is widely discussed, for instance in [24]. Autonomy, reliability, and flexibility are the most common requirements of such networks [25].

- **Industrial infrastructures applications**

Reliability, availability, maintainability, and mobility characteristics are crucial for the WSNs implemented in industrial infrastructures and automation applications. Tasks of events detection, periodic data collection, real-time data acquisition, equipment control, robots control and real-time inventory management have been assigned to WSNs. WSNs implementation has some advantages in industrial infrastructures compared to the wire communications. The benefits include flexibility of installation and upgrading the network, lower costs of deployment and maintenance, decentralization of tasks automation, flexibility for moving and rotating devices, easier fault diagnosis. Possible interfaces to wide area networks from different networks can help improving the efficiency of automation infrastructures. An additional advantage is the high interconnection capability of integrated wireless sensors with built-in communication using micro-electromechanical systems (MEMS) [26].

- **Military applications**

WSNs is one main part of the modern military logistics, operations, and human resources protection. The requirement and challenges of the WSNs design depend on the operating scenarios for which they are intended. Most of the military applications are large-scale implementation in which the WSNs are non-manually deployed. Different sensor types are implemented in particular WSNs for supporting defense strategies, environment surveillances, and logistics support [27].

- **Medical and healthcare applications**

High performance and reliability computational devices, smart sensors, and high sensitive measurements devices are required in WSNs architecture implemented in healthcare so as to allow in-home assistance or telemedicine and perform patient progress monitoring and emergency situations. Also, light or voice reminders could be used for the patient to remember the medical data and time [28].

- **Transportation and mobile applications**

The efficient traffic management systems are required to cope with the rapid increasing of traffics around the world, with accidents avoidance support. The WSNs used in intelligent transportation systems have introduced new ideas about the smart city applications with the capability to offer traffic safety and congestion control [29].

- **Internet of things**

Internet of things (IoT) is the capability of physical objects or things with embedded smart system to communicate and sense their internal and external environment. The WSNs are used to provide the communication infrastructures for the IoT to introduce new applications, services, and the smart world. Figure 1.7 explains the IoT applications and future [6].

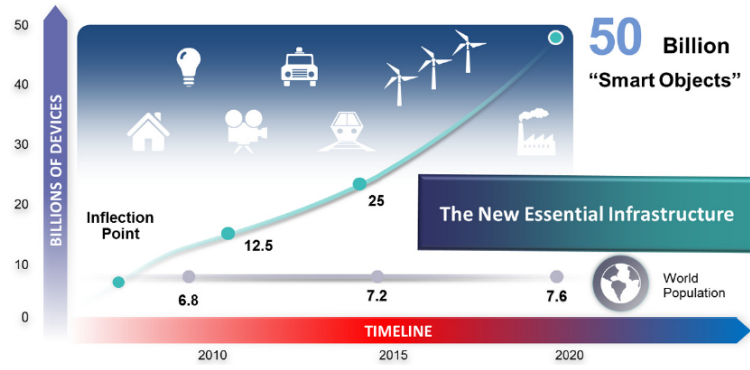


FIGURE 1.7: IoT future and applications [6].

Several applications and research challenges related to IoT are brought out and classified by Miorandi et al [7] as in figure 1.8 [7].

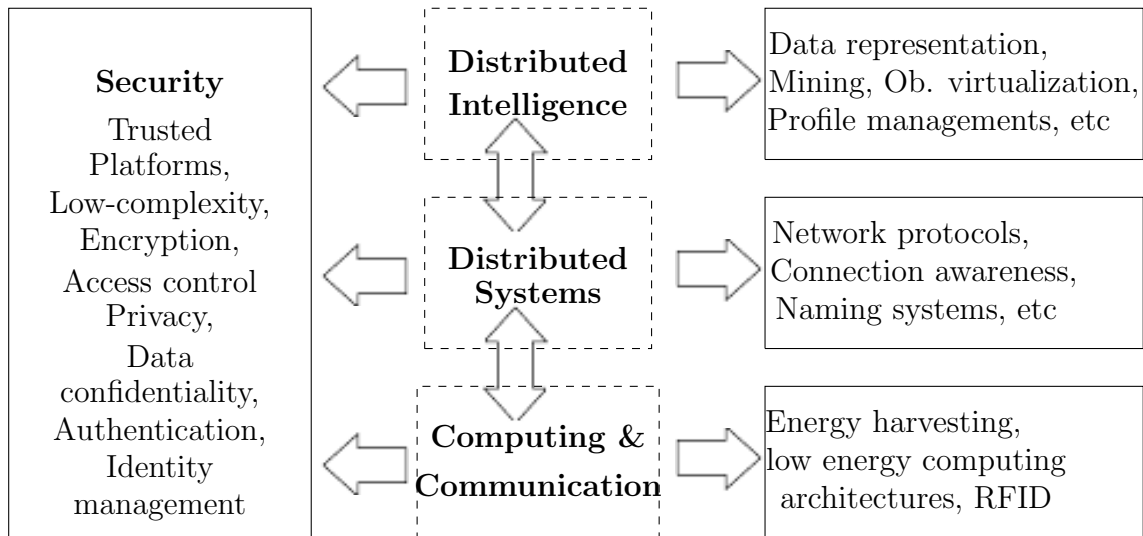


FIGURE 1.8: IoT research challenges [7].

• Cyber physical systems

The Cyber-physical systems (CPS) are systems that integrate natural and human-made physical systems with computation, communication, and cybernetics as an interaction between physical and computational environments. The sensing and sensor networks are the provider of the communication and data collection for the cyber-physical systems and its applications [30] [31].

The characteristics and advantages of WSNs are still attractive for new applications to be introduced. This large area of applications has brought out more research challenges to be investigated in WSNs. The next paragraph introduces some of them.

1.2.2 Wireless Sensor Networks research challenges

The evolution of WSNs, which merged many fields with a broad scope of applications and brought out many attractive issues, are addressed by intensive investigations. The problems include localization, connectivity, coverage, obstacle adaptability, node density, communication and sensing range, energy, lifetime, sensor relocation, movement of sensors, fault tolerance, reliability, ...[32] [17]. The next paragraphs present some of the WSNs research challenges.

- **Deployment**

For localization and deployment, the targeted object or field position should be known. Random deployment could be used by default to access environments. The sensor node could be efficiently used or not completely used according to its position [33]. Network cost, coverage, connectivity and lifetime constraints could be considered to optimize the WSNs deployment [34] in addition to the targets mobility and tracking [35].

- **Connectivity**

Many protocols are developed recently as connectivity and routing strategies considering the dynamic topology of the WSNs to ensure the collected data propagation to the end destinations. For more details about routing protocols of mobile ad-hoc Networks, the reader can refer to [20] for example.

- **Coverage control**

Many protocols were developed to providing a continuous and effective coverage for the area of interest or the region to be monitored as one of the

important research challenges faced in the WSNs [36]. A centralized control and distributed control are the main visible strategies used in coverage control developed for the ad-hoc sensor network [22].

- **Self-adaptivity**

The self-adaptivity and the adaptability to obstacle typically come together when describing the WSNs behavior or responding to the environmental changes or barrier. The capability of the WSNs to reorganize itself with the new situation or adaptability ought to be considered in the WSNs deployment and installation [37] that could support the fault-tolerance for WSNs based systems [38].

- **Nodes density**

The node density is used to describe the number of nodes allocated in the targeted field. It has a direct effect on all or most of the WSNs research challenges such as the connectivity, energy consumption, adaptability, resiliency and lifetime. Considering that the node has the capability to act as sender and router, the power consumption in each node is affected by its position, the nodes near the base station spend more energy for data routing. Therefore, the node density should be greater in the neighborhood of a base station, or the sensing nodes will not be connected soon and so the node lifetime will be shorter [39].

- **Communication range**

For each node, the communication range is defined as the circle area “or section” inside which the neighbor nodes can receive its signal with a radius equal to r [40].

- **The energy**

The power management [41] is always considered in WSNs design regarding the available energy amount used for sensor node operations. Typically, the sources of energy are batteries which must be replaced or recharged after the drain. For some nodes, both options are not applicable. Therefore, the sensor nodes will just be discarded once their energy reserve is depleted [15].

The efficient coverage control methods could optimize the energy utilization, the WSNs lifetime [42] and the number of nodes required for coverage [43].

The energy reserve, coverage control, node density, deployment are main factors to prolong the WSNs lifetime, availability and reliability. How can these factors affect the lifetime is described in details in the next chapter as the main problem of this work.

1.2.3 Wireless Sensor Networks description and modeling

For every WSN, a limited number of sensor nodes are deployed in two-dimensional or three-dimensional space, for the surveillance of specific targeted objects or fields. The sensor nodes send the collected data to static or mobile base stations. Therefore, the deployment of the sensor node requires to specify the values of (x, y) of each node in the 2-dimensional space or (x, y, z) for 3 dimensions. Then, the capability of each sensor node to monitor all or part of targeted objects or fields mainly depend on this deployment and the coverage range r . Every sensor node i can monitor every target j , if the distance d_{ij} is less than or equal r . Figure 1.9 displays 10 sensor nodes deployed in 10x10 area with $r = 1.5$.

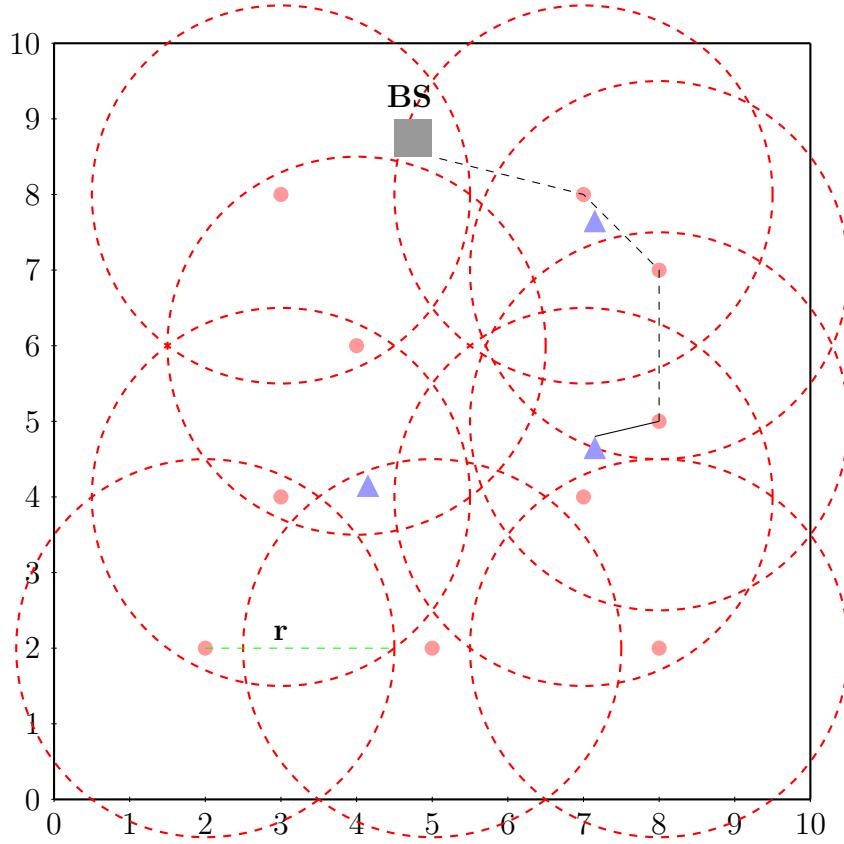


FIGURE 1.9: WSNs deployment.

The connectivity and data transmission between sensing nodes also depend on deployment values (x, y) of the sender and receiver in addition to the positioning of the base station (BS). The distance between sender and receiver d_{sr} must be less than or equal to r to ensure a possible connectivity to the BS.

1.2.4 WSNs lifetime optimization through energy consumption

The WSNs lifetime, as one of the most significant design challenges [44], is exactly the time elapses until all available sensors couldn't satisfy the required coverage constraint. The lifetime may be affected by different factors such as sensor degradation or failure and the available amount of energy. When the reserve of energy is not renewable, and the sensors is operating in some critical area or location [45], the reserve of energy become the most important factor. Therefore, the lifetime

of the WSNs could be optimized through optimization of energy consumption. In ad-hoc WSNs, the energy consumption depends on the data sensing rate, receiving and transmission rate[46] [47], which depend on the nearness of the node to the base station. Although there are many recent works regarding this problem, it has not been fittingly solved [48].

1.2.5 WSNs reliability, failure and self-adaptivity

Reliability is one of the most significant requirements for a wireless sensor networks applications in the areas of industry, healthcare, and environments. Reliability level of the network can be evaluated using reliability modeling and analysis as key steps for designing and optimization of sensor network systems. Whatever the strategy used for sending the sensed data to the end user, data cannot be delivered if the path fails, which may happen either in the communication link or the WSN node. A link failure can happen due to different factors such as noise, interference, distance, or environmental conditions. While, the WSN node can fail due to firmware factors (embedded operating system) or hardware factors (radio, sensors and energy devices) failures [49]. In [49] the reliability of a node is defined as sequential sorted blocks of all factors such as application, firmware, middleware, hardware, radio communication or battery level. As the reliability of a WSN node is a function of the reliability of its components arranged in series, if one of them fails, the whole node fails. Regarding the reliability, the WSNs design, and deployment [50] should consider sensor node constraints like battery power, transmission range, sensing range and processor capability. The energy, sensing, processing and communication issues should meet the reliability requirements, and the communication ought to consider the connectivity requirements so that the end user can access the network and receive the expected data sensed and processed by the sensing nodes. Achieving the overall network reliability of the communication process is to construct a network with the minimum number of reliable links and each link must be feasible [51]. The self-adaptivity has different definition[52]: it is the capability of the system to adapt its behavior according

to the environment or the ability of a system to achieve its goals in a changing environment, by selectively executing and switching between operating models. Therefore, a self-adaptive system evaluates its behavior and changes its operation when the evaluation indicates that its performance is not sufficient. Finding a better possible configuration or performance according to the most common stages is depicted in figure 1.10. The tasks for self-adaptivity includes: monitoring of the targeted system or the environment to collect the conditions data required for adapt, analysis of the collected data to make adaptive decisions in the next-step, determine and plan the steps to achieve adaptivity, and execute the steps.

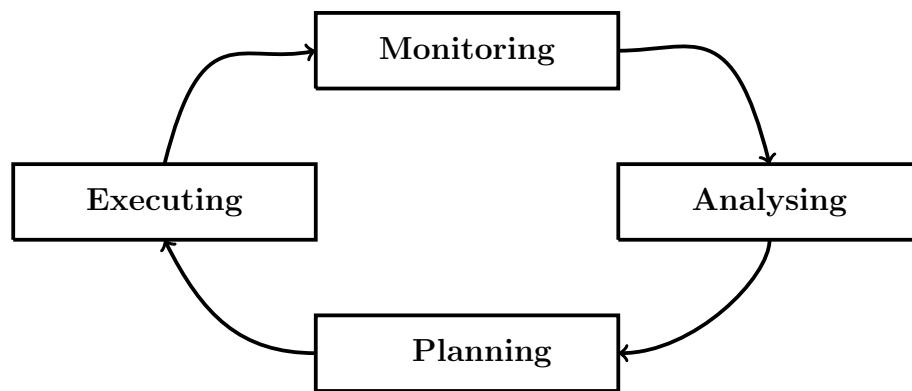


FIGURE 1.10: The self-adaptivity common stages

At all time, the system components are monitored, the collected data analyzed and estimated to the required, and then the next operation strategy is planned “if available” and executed.

The WSNs lifetime should be optimized considering the limited energy reserve, the component degradation and the probability of failure constraints. An efficient resources utilization, tasks assigning and scheduling method could prolong the WSNs lifetime. The next subsection describes the scheduling environment, constraints, objectives and its applicability in WSNs.

1.3 Scheduling problem

Planning and scheduling as defined in [53] are decision-making processes that are used on a regular basis in manufacturing and services. Mathematical techniques

are utilized in all planning and scheduling functions. Solving a scheduling problem can consist to organizing a set of activities (jobs or tasks) to be executed, by using the available resources capacities. This execution has to consider and follow different technical rules (constraints) to achieve the maximum efficiency of the resources (according to a set of criteria or objectives) [54]. The number of jobs and the number of machines are assumed to be finite and denoted by n and m respectively. Then the pair (i, j) refers to the processing step or execution of job j on machine i . The following pieces of data are associated with job j [55]. The task index, processing time, release time and priority index are relevant variables in scheduling problem.

1.3.1 Modeling the scheduling problems in WSNs

The problem of scheduling in WSNs could be considered as follows:

Given a set of sensors and a set of targets, find the sensors assignment to targets coverage that maximizes the total coverage time or more objectives. Figure 1.11 gives a suggestion of 5 sensors assigned to cover 4 targets.

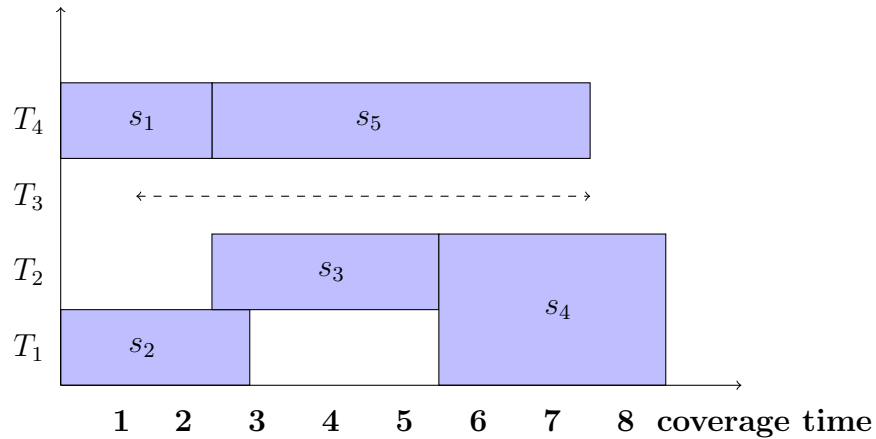


FIGURE 1.11: Sensor to targets coverage scheduling

In figure 1.11, T_3 is not covered through all coverage time while T_1 and T_2 are not covered for a part of the coverage time. In most cases, the scheduling part required after covers determining is left without solution [56] [57].

To ensure that all targets are covered all time, the possible number of covers should be found "not necessary to be disjoint." Then, one should search for covers to targets assignment to optimize the coverage time. Figure 1.12 gives a suggestion for 4 covers composed of 7 sensors assigned to cover 4 targets.

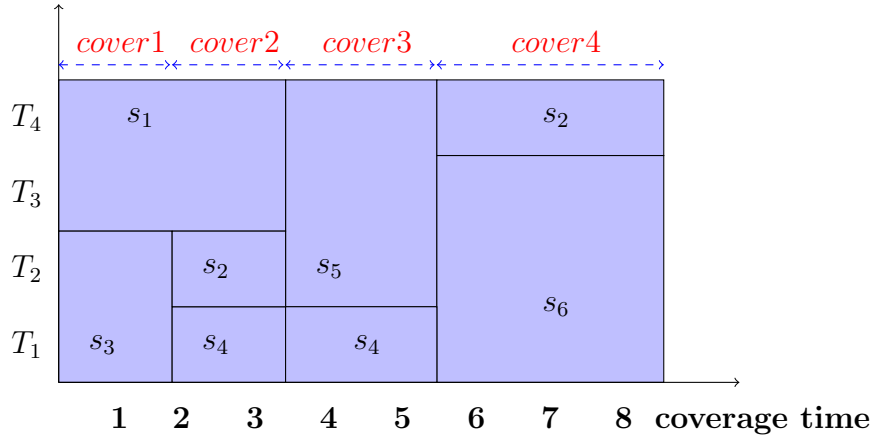


FIGURE 1.12: Covers to targets coverage scheduling

In figure 1.12, cover 1 includes S_1 and S_3 , cover 2 includes S_1 , S_2 and S_4 , cover 3 includes S_4 and S_5 and cover 4 includes S_6 and S_7 . Therefore, this problem could be split into two sub-problems:

1. Given a set of sensors and set of targets, find the optimal covers.
2. Given a set of covers and a set of targets, find the coverage assignment that optimizes the coverage time.

In addition to the coverage, the parallel processing running on the collection of interconnected sensor nodes to execute a set of processes could be modeled as parallel machines, (see [58] for more details).

1.3.2 Scheduling environments, constraints, and objectives

The scheduling problem description is always composed of three parameters α, β and γ . The α field describes the machine and resources environment and contains

generally one entry. The β field provides the necessary details about the processing constraints, and it can contain a single entry, multiple entries or nothing. The γ field describes the objective to be minimized and contains, in general, a single entry [55]. The possible situations of the machine environments specified in the α field are: 1) single machine (1) as the simplest of machine environment, algorithms such as weighted short processing time first (WSPT), weighted short discounted processing time (WSDPT) and earlier due date first (EDD) can be used to achieve the objectives, 2) parallel machines models in which a set of machines in parallel, widely applied in information systems, can be: identical machines in parallel (Pm), machines in parallel with different speeds (Qm), or unrelated machines in parallel (Rm), 3) flow shop (Fm) is a set of machines in series, each job has to be processed on each one of the machines. It could be generalized as flexible flow shop (FFc) when identical machines in parallel used in a subset of the series stages. 4) the job shop (Jm) in which each job has its required route to follow in the environment, it could be generalized as a flexible job shop (FJc) when identical machines in parallel are used in a subset of the series stages. 5) the open shop (Om) determines a route for each job, different jobs could have different routes is allowed, some of the processing times on each one of the machines may be zero. The restrictions and constraints that could be found in β field includes: release dates, sequence dependent, preemption ($prmp$), precedence constraints ($prec$), representation as a directed acyclic graph (DAG), machine eligibility restrictions (Mj), permutation ($prmu$), blocking ($block$), no ? wait (nwt) and recirculation ($recirc$).

Regarding the input variables of every scheduling problem (the environments (resources) variables and constraints), the scheduling problems aims to perform the following possible objective functions always to be minimized in the γ field [55]: 1) the makespan ($Cmax$) that defined as $max(C_1, \dots, C_n)$, is equivalent to the time required for the last task to leave the system. Many approximation algorithms developed for finding the minimum makespan of single machine [59] or parallel environments. A minimum makespan usually refers to a good utilization of the resources, 2) the maximum Lateness ($Lmax$) that defined as $max(L_1, \dots, L_n)$. It measures the worst violation of the due dates. This objective has been studied for

different applications and constraints through many algorithms as in [60], 3) total weighted completion time ($\sum w_j C_j$) which is the sum of the weighted completion times of the n jobs gives an indication of the total costs to be incurred by the schedule. The total weighted completion time or the weighted flow time is aimed to be minimized in many information system applications such as data centers [61], 4) the discounted total weighted completion time ($\sum w_j(1 - e^{-rC_j})$) is a more general cost function compared to the previous one, where costs are discounted at a rate of r , $0 < r < 1$, per unit time. That is, if job j is not completed by time t , an additional cost $w_j r e^{-rt} dt$ is incurred over the period $[t, t + dt]$. If job j is completed at time t the total cost incurred over the period $[0, t]$ is $w_j(1 - e^{-rt})$. The value of r is usually close to 0, say 0.1 or 10%. 5) total weighted tardiness ($\sum w_j T_j$) is also a more general cost function than the total weighted completion time, 6) weighted number of tardy jobs ($\sum w_j U_j$) is not only a measure of academic interest; it is often an objective in practice as it is a measure that can be recorded very easily. All the objective functions above are so-called regular performance measures.

1.3.3 Complexity of the scheduling and optimization problems

Solving a scheduling and optimization problem amount to finding the optimal or near-optimal solutions for the objective function considering some goals and restrictions. The complexity of the optimization problem could be evaluated based on the computational resources required to solve it considering both time and space complexity. The optimization problems are classified in different groups or complexity classes according to the computational efforts required to find the optimal solution. The problems in one complexity class could have limits linked to the computational complexity, which depends on the size n of the problem or its input size. the complexity classes includes the polynomial P , non-deterministic polynomial NP , NP-complete and NP-hard [62] The complexity P class is a set of optimization problems that can be solved in polynomial time complexity in the worst-case. The time required for solving effectively this problem in P is bounded

for any instance of the problem with n inputs ($n > 0$) by a polynomial function of the type $O(n^k)$. The complexity NP class includes problems with practical importance. It describes the set of optimization problems that can be solved in polynomial time in worst-case using a non-deterministic algorithm. The NP – *complete* belongs to NP , and there exist polynomial algorithms to transform every problem in NP into it. The NP – *hard* is not in NP but there is a NP – *complete* problem that can be transformed into it with a polynomial time. It is assumed that P is a subset of NP ($P < NP$) whether some problems are in NP , but not in P . The Complexity classes relations are depicted in figure 1.13 [63] [64].

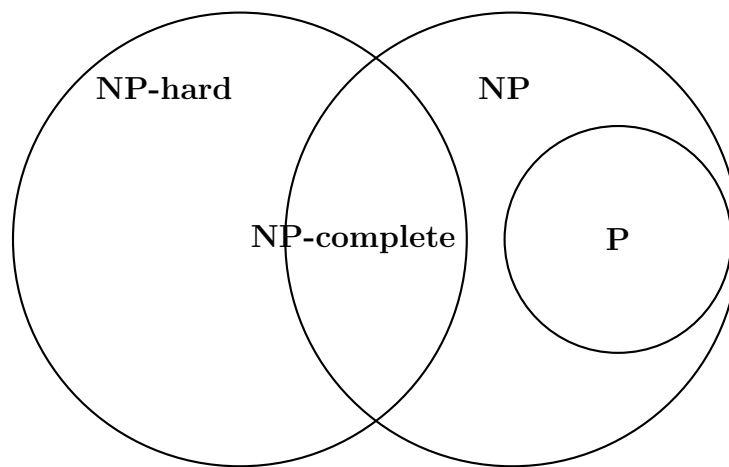


FIGURE 1.13: Complexity classes.

1.3.4 Scheduling classes

In scheduling terminology, a sequence usually corresponds to the n task permutation or the order of jobs processing on a given machine while the schedule usually refers to an allocation of tasks within a more complicated setting of machines that allows the possibly for preemptions of tasks by other tasks that may be released at a later time. Different scheduling classes with different operating conditions could be abstracted as in figure 1.14.

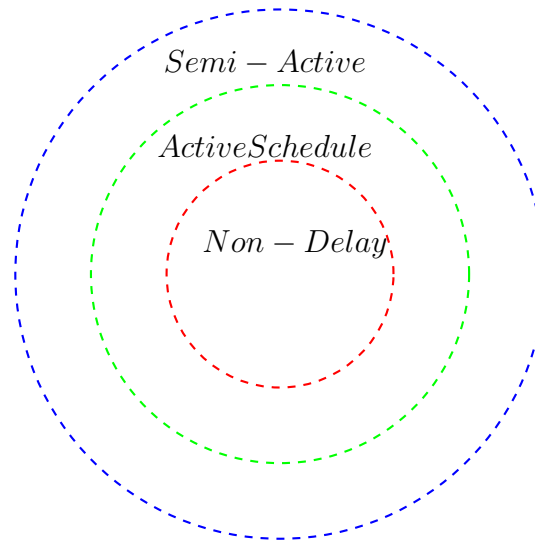


FIGURE 1.14: Scheduling classes.

A feasible schedule is called non-delay if no machine is kept idle while an operation is waiting for processing. Requiring a schedule to be non-delay is equivalent to preventing unforced idleness. A feasible non-preemptive schedule is called active if there is no possibility to construct another schedule, by changing the order of processing by the resources, with at least one operation finishing earlier and no operation finishing later. A feasible non-preemptive schedule is called semi-active if no operation can be completed earlier without changing the order of processing on any one of the resources.

1.4 Optimization approaches

Optimization problems are common in many fields and different domains in the human activities where we have to find an optimal or near-optimal solutions for specific problems with the capability to meet some limitations. The most common optimization problems characteristics include the following:

- 1) It has many alternatives of decision and possibilities of solution.
- 2) Additional constraints can limit and decrease this number of available alternatives.
- 3) Each decision can generate a different effect on the evaluation criteria.

4) It has an evaluation function that based on these alternatives described as function of the decision variables [63].

Given a set of decision variables $V = \{v_1, v_2, \dots, v_v\}$, optimization considered to obtain the best solution for an objective function of this decision variables $f(V)$ according to some restrictions on the decision variables. The best solution is obtained either by minimizing or by maximizing the objective function, and the optimization consists of finding the condition of the decision variables that gives the maximum or minimum value of the objective function. Figure 1.15 explains that if a point $v_i \in V$ is approved with in the minimum value of the function $f(v)$, the same point is also confirmed in the maximum value of the opposite of the function, $-f(x)$ [65] .

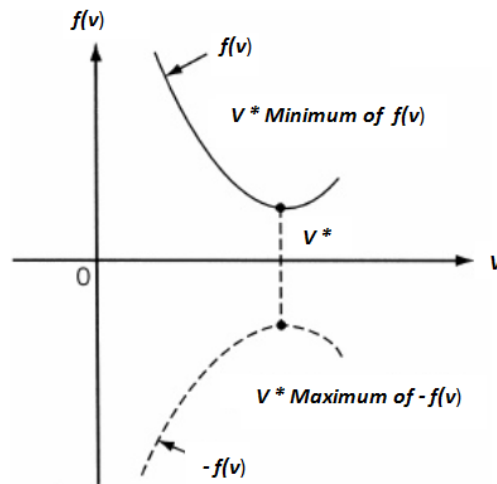


FIGURE 1.15: Maximization and minimization of $f(v)$.

The optimization problem, in general, has the following mathematical formulation:

- The objective function

$$\text{Maximize/Minimize } f(v_1, v_2, \dots, v_n)$$

- The constraints

$$\Phi_i(v_1, v_2, \dots, v_n) \leq 0 (i = 1, \dots, l)$$

$$\varphi_j(v_1, v_2, \dots, v_n) \leq 0 (i = 1, \dots, m)$$

This formulation is normally referred to as the general nonlinear programming problem. The feasible point of a solution is any point value of the vector V that satisfies all these equations [66].

1.4.1 Combinatorial optimization

Combinatorial optimization methods are useful in a particular type of mathematical optimization problem in which the set of feasible solutions of the problem is finite. Such a problem is defined, in its most general form, on a finite set of feasible solutions with a reasonable characterization [67]. The computational problems normally require significant efforts to search a huge number of candidates for the optimal solution in which the evolutionary algorithms could be used. The evolutionary algorithms are natural principles based computational methods developed as a simulation of natural behavior to be implemented in computer science for human systems developing. This kind of algorithms is involved in many fields of research, development, and applications as in figure 1.16 [64].

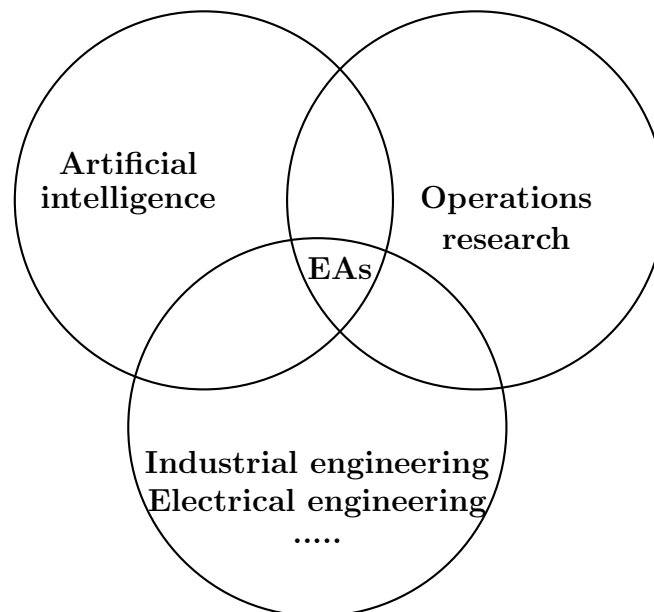


FIGURE 1.16: Evolutionary algorithm applications.

The key features of the evolutionary algorithms include: 1) has group of solutions or individuals to be enhanced called Population (Population-based); 2) the solutions or individuals in a population have its value or representation (code), and the evaluation of this values is called its fitness value (Fitness-oriented); and 3) Variation-driven.

1.4.2 Optimization methods

This subsection aims to describe the methods, techniques or strategies used to solve the optimization problems in many fields of application. According to the quality of solution guarantee, these could be classified in exact and approximation or heuristics as in the next paragraphs.

1.4.2.1 Exact methods

This sub-section describes the exact methods that aimed to guarantee the optimal solution, even if it takes greater computational efforts regarding the resources and time. The exact methods include the Branch and Bound, the Branch and cut, the Simplex, etc. Several examples are described below in addition to the linear programming used in this work.

Linear Programming The common form of the linear programming is as follows: *Maximize*

$$f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 \dots + c_nx_n \quad (1.1)$$

Subject to :

$$\sum_{i=1}^n a_{ij}x_i \leq b_j \forall j = 1, \dots, m \quad (1.2)$$

$$0 \leq x_i \forall i = 1, \dots, n \quad (1.3)$$

where m and n are given natural numbers, c_i, b_j and a_{ij} are constants and x_i are decision variables. Expression (1.1) is the objective function to be maximized or minimized and expressions (1.2) and (1.3) are constraints. With one more characteristic that both the objective function and the constraints are linear equations or inequalities [68] [69]. For a linear programming problem with two variables as the simplest case, the optimal solution can be obtained by using a graphical method as in figure 1.17

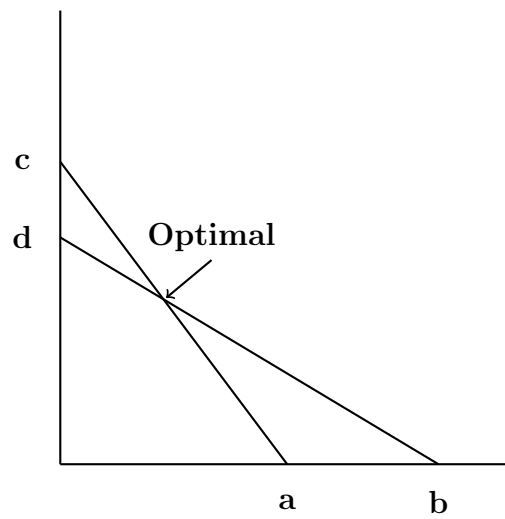


FIGURE 1.17: Graphical method for simple LP - unique solution.

In some cases, the optimum solution may not be unique for example in the case of parallel function as in figure 1.18.

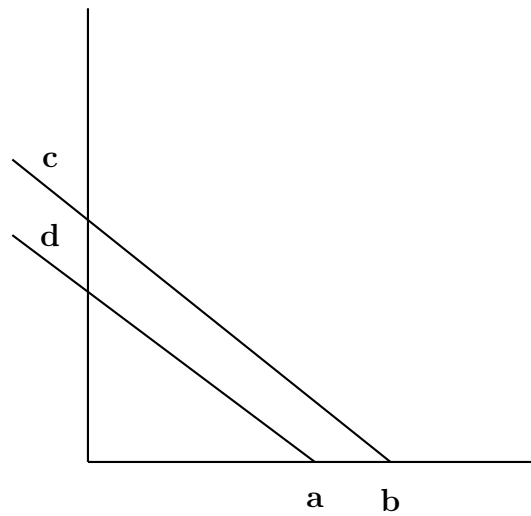


FIGURE 1.18: Graphical method for simple LP - parallel function.

Branch and Bound The branch and bound method is one of the main strategies used for solving discrete and combinatorial optimization problems. Regarding a combinatorial optimization problem a finite set of feasible solutions, the branch-and-bound deals with these feasible solutions in a systematic manner to find the optimal solution of the problem. It tries to solve the combinatorial problem by dividing it into smaller problems and computing an upper and lower bounds for each of the smaller problems that may be employed to exclude parts of the solution set out of consideration [67]. The branch and bound method has three main steps: 1) selection, 2) branching and 3) bound, with an appropriate rule or function should be defined for each step. See [70] for applied branch and bound example.

1.4.2.2 Heuristics and meta-heuristics

For hard problems, the exact algorithm that can guarantee the global optimal solution within an acceptable time might not be possible. Thus many heuristic algorithms have been developed for finding faster near-optimal solutions. Heuristic algorithms can quickly generate a solution with acceptable quality. But there is no guarantee for an optimal solution can be obtained and the time to derive a solution is also long in some worst cases [64]. Recent years have brought out a significant growth in the development of heuristic procedures to solve optimization

problems. There are many motivations and reasons for using heuristic methods. 1) No method to resolve the problem to optimality is known; even if there exists an exact method to solve the problem, it cannot be implementable on the available computing hardware. 2) The flexibility of the heuristic methods compared to the exact methods. 3) The heuristic method could be used as part of a global procedure that aims to find the optimum solution to the problem. The good heuristic algorithms should have the following characteristics: 1) It can obtain the solution within reasonable computational effort; 2) The quality of solution should be high and should have a high probability to find the near optimal solution; 3) The probability of obtaining a far from optimal or a bad solution should be very low. The heuristics and meta-heuristics include the Genetic Algorithms, the Greedy Algorithms, the Simulated Annealing, etc. [67]. It is important to highlight some of them in the next subsection, especially the genetic algorithm that we developed and implemented in this work.

Genetic Algorithm It is common in computer science to search for a feasible and acceptable solution of a decision support problem among a collection of candidate solutions. The genetic algorithm (GA) is an approach to solution search introduced by Holland in the 1960s [14]. Compared to the evolutionary programming strategies, Holland's goal was not only to design algorithms to solve complex problems but also to study and develop methods inspired by the natural adaptation processes applied to computing. Thus, the GAs have become the most modern evolutionary computation research technique. All the genetic information in humans is stored in 23 pairs of chromosomes. Each of these chromosomes is composed of several parts called genes as in figure 1.19. The genes code the properties and the characteristics of an individual and determine the characteristics of the next generations, an interesting aspect of evolution [71].

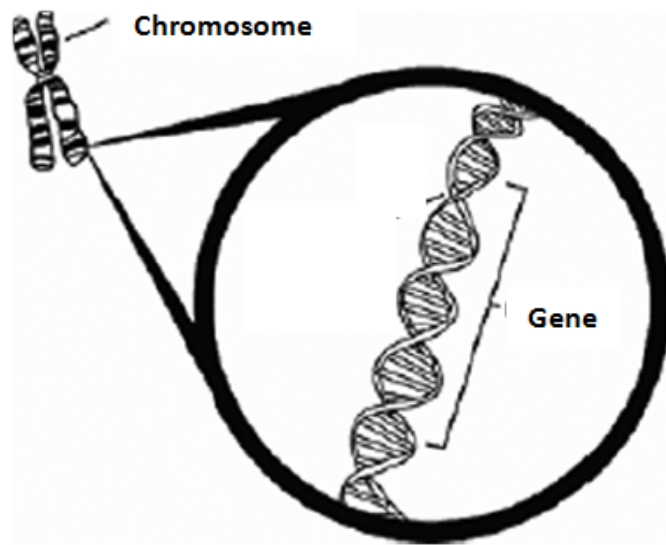


FIGURE 1.19: Genes and chromosomes

The GA simply executes a number of iterations or generations of selection, modification and update of a set of candidate solutions or population [72] as basic evolution cycle in figure 1.20

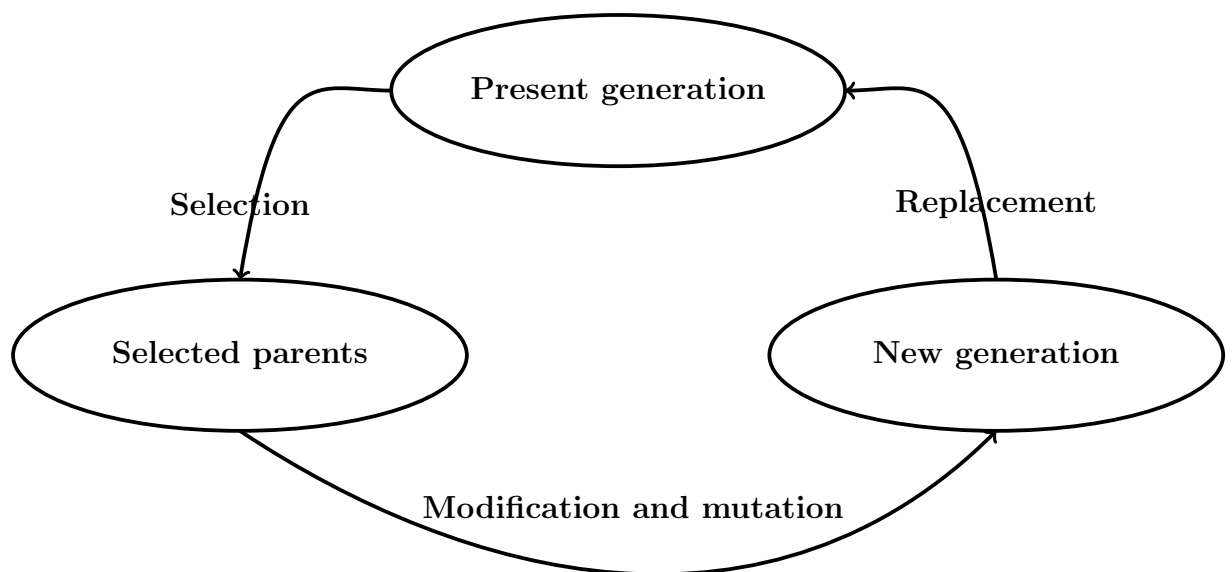


FIGURE 1.20: A simple evolution cycle.

The principles of a simple genetic algorithm is an integration of terminologies (subfunction and steps) includes the encoding, populations, individuals, crossover,

mutation, selection, fitness, evaluation and iterations that are used in a specific way to formulate a GA based method [73].

- **The encoding process:**

This is used to represent the solution into individual genes and chromosomes so as to be processed using GA operators and functions. It can be performed using bits (binary), octal encoding, numbers (integer, real), arrays or any other objects.

- **Population:**

A population is a set of individuals or candidate solutions. The initial population generation consists of a given number of individuals equal to the population size. This initial population is usually created randomly in GAs to be used as starting point of the searching process.

- **Fitness and selection:**

The fitness function is used in genetic algorithms to calculate the value of the objective function or constraints for its individual. To calculate the fitness, the chromosome has to be decoded first, then use the objective function for the evaluation. The assessment gives an indicator, which corresponds to how the chromosome is close to the optimal solution. The purpose of the selection process is to find fitter individuals of the population to be used as parents for reproduction of the next generation.

- **Crossover:**

The crossover is the reproduction operator used for producing new children based on the selected parents. The parent's offspring is typically composed of three steps: selects a pair of individual strings as parents, select the crossing at random along the chromosome length and finally, exchange the position values between the two strings according to the crossing points. Different types of crossover are developed such as single point crossover, two-point crossover, N-point crossover, uniform crossover, three parent crossover, ordered crossover, partially matched crossover, etc. The single point crossover and two point crossover are depicted in figure [1.21](#)

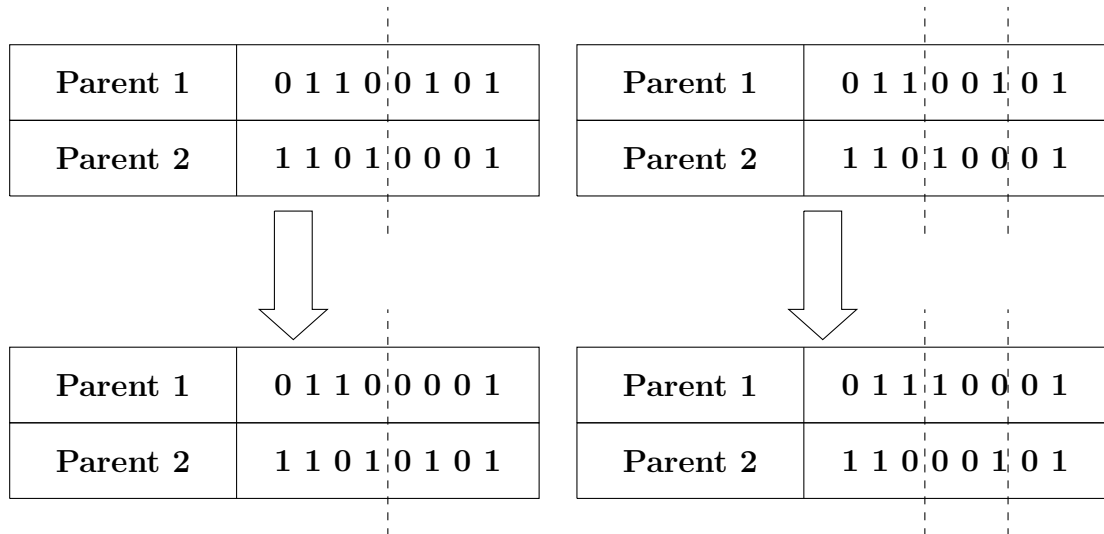


FIGURE 1.21: Single point crossover and two point crossover

For the partially matched crossover, for two with the same length, two crossover points are selected randomly and uniformly along the length. These crossover points give a possibility for a matching selection used to affect a crossover operator through position exchanges as in figure 1.22

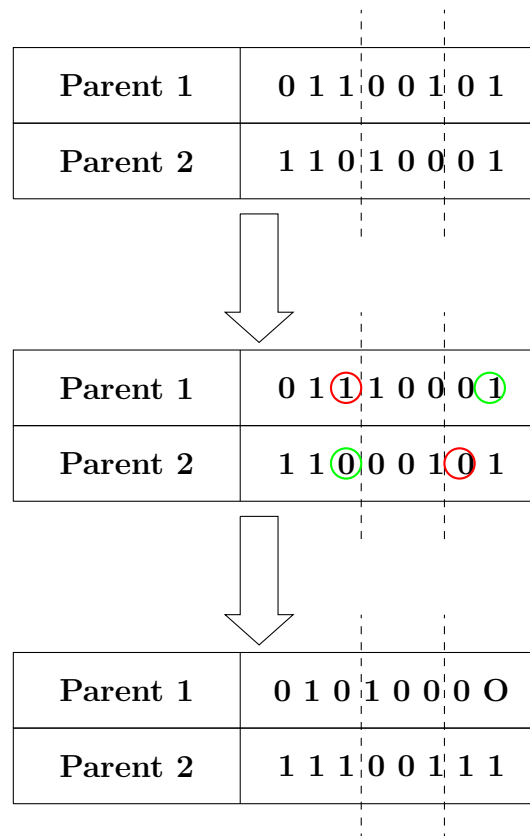


FIGURE 1.22: The partially matched crossover

- **Mutation:**

The mutation is used for preventing the algorithm from being trapped in a local optimal search. Different forms of mutation according to the various types of representation such as: reversing, interchanging, replacement, etc. [71]. The mutation can be done in general in two steps: genes selection and genes modification. The genes selection process is typically random, and many different modification functions are used according to problem nature and coding.

Many versions of GAs were developed such as parallel distributed GA, fine-grained parallel GAs (cellular GAs), multiple-deme parallel GAs, hybrid GA, adaptive AG, fast messy GA and independent sampling GA [71]. The principles of a simple genetic algorithm is depicted in figure 1.23.

Many heuristics were introduced and implemented in the literature including methods such as, Greedy Algorithm, Ant Colony, and simulated annealing for example.

Simulated Annealing: This method was imported from the physical annealing process of heating up and cooling down solid metals. The method was defined in combinatorial optimization and developed to finding a solution with a minimal cost from a large number of solutions. Thus, the simulated annealing (SA, for short) is a method designed for solving problems in the field of combinatorial optimization, by simulating the physical annealing process. The SA has many features such as the possibility of finding a high-quality solution, simple mathematical modeling and not need large computer memory. Furthermore, it is possible to start the SA with any given solution and try to enhance it which could be used to improve a solution obtained by other heuristic methods [66]. The exact, as well as the heuristic methods, have been suggested by researchers and implemented for solving the WSNs lifetime optimization problems, as will be presented in the next chapter.

Considering that the heuristics could not grantee the optimal solutions, there are many possibilities for the quality assessment of the heuristic base solutions. The

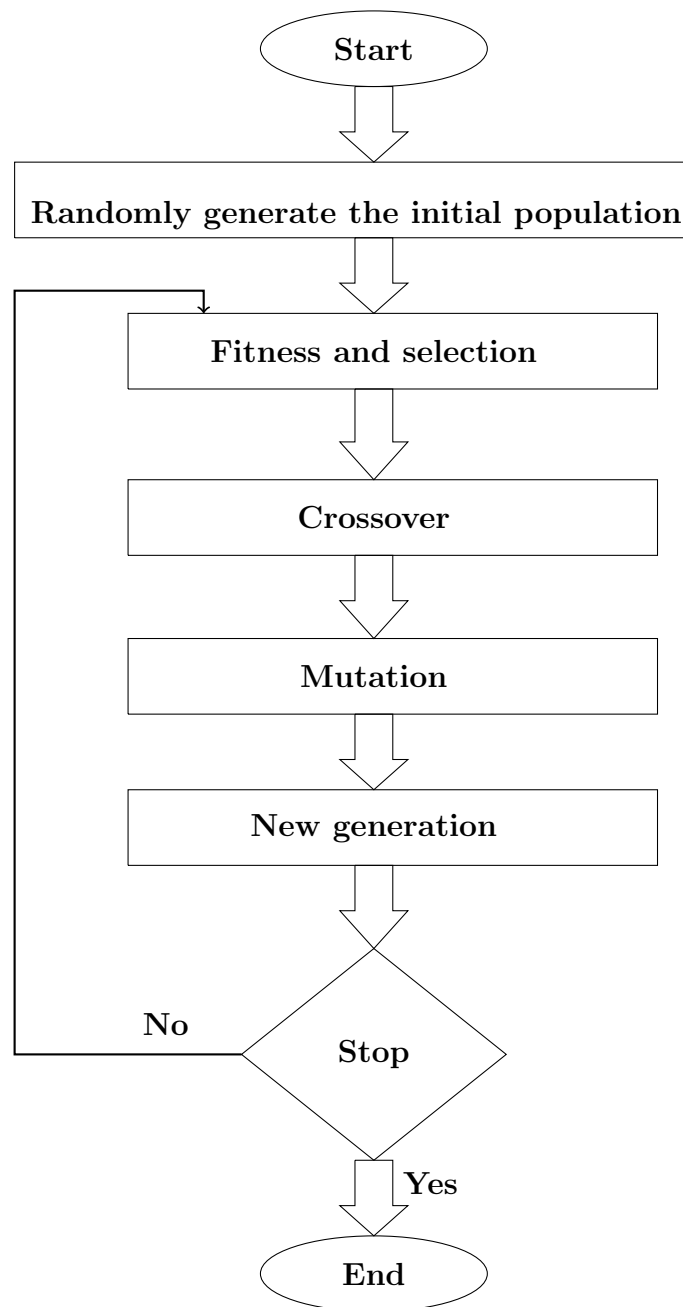


FIGURE 1.23: The simple genetic algorithm

later could be compared to an optimal solution, to an appropriate bound, or to other heuristics solutions [74]

1.5 Conclusion

Regarding the WSNs research challenges and the capabilities of the scheduling models and optimization techniques, the WSN is one of the most recent popular fields of scheduling and optimization application. The optimization methods are widely implemented to solve a WSNs research problem. The optimal deployment [75], data aggregation [76], optimal mobile sink node positioning [77] and lifetime optimization [77] are only some examples of the optimization methods application to WSNs research challenges. The problem of WSNs lifetime is a scheduling and optimization problem as described in many research work [78], especially, when a limited number of sensing node with a limited energy reserve are used. Limited resources "sensor nodes" are aimed to be optimally utilized to achieve an optimal coverage period "lifetime". The lifetime modelizes the objective function to be optimized, and the restrictions on the resources utilization stand for the constraints. This kind of limited resources problems is suitably modeled by LP if it is possible to model both the objective function and the constraints with linear equations or inequalities. The performance and efficiency of GAs put it as the highest priority to be used for solving such kind of problems within reasonable computational efforts compared to the other evolutionary strategies. This work is aimed to formulate the problem of WSNs lifetime optimization considering its limited energy resources into ILP mathematical model and find the optimal solution of this problem. The ILP optimization tools are used to find the exact solution and the GA-based algorithms with different configurations of crossover and mutation operators are used to search for near-optimal solutions in reasonable computational efforts. More description about WSNs lifetime formulations and the methods used to solve this problem in the literature are mentioned in the next chapter.

Chapter 2

Problem Statement as Disjoint Set Covers Maximization

2.1 Overview

This chapter describes the problem of WSNs lifetime optimization considering the limited initial energy reserve. First, it explores the previous works on this issue, the definitions, and theorems required for a good understanding of this problem. The WSNs lifetime optimization problem is widely addressed in the recent years. In a significant amount of the research work, this problem has been formulated as a disjoint set covers (DSC) maximization problem, as mentioned in the next subsection. Other researchers have preferred to formulate this problem by a mathematical model, then, they have suggested exact and heuristic methods to find an optimal or near-optimal solution. We prefer to review this works in the following subsections considering the Problem statement declaration and the methods used.

2.2 Problem statements and description

This section explains the problem of lifetime optimization in WSNs and its related basic definition. Also, it presents the utilization of DSC maximization, and the latter was used in WSNs lifetime optimization. Then, it describes how can the WSNs lifetime be formulated as a scheduling problem, and finally, it gives an overview of the objective function to maximize and the constraints to be considered. Table 2.1 describes the notations used to formulate the problem for a set $S = \{s_1, s_2, \dots, s_m\}$ of sensors used to monitor a set $T = \{t_1, t_2, \dots, t_n\}$ of targets.

TABLE 2.1: Problem notations

S	Set of sensors $\{s_1, s_2, \dots, s_m\}$
T	Set of targets $\{t_1, t_2, \dots, t_n\}$
m	$\text{Card}(S)$ = number of sensors in S
n	$\text{Card}(T)$ = number of targets in T
i	Sensors index $\subset S$, $i \in \{1, 2, \dots, m\}$
j	Targets index $\subset T$, $j \in \{1, 2, \dots, n\}$
$S(t_j)$	Set of sensors that can cover target j
$T(s_i)$	Set of targets that sensor i can cover
E_i	Initial energy of sensor i
C	A collection of subsets of sensors that can cover all the targets in T
q	$\text{Card}(C)$ = number of covers in C
l	Covers index $\subset C$, $l \in \{1, 2, 3, \dots, q\}$
$S(C_l)$	Set of sensors included in cover C_l
$T(C_l)$	Set of targets covered by cover C_l
L	Network lifetime

Each sensor $s_i \in S$ has the capability to cover a subset of targets $T(s_i) \subset T$. $T(s_i)$ could be all or part of T , $\text{Card}(T(s_i)) \leq n$, and each target $t_j \in T$ could be covered by a set of sensors from S , $S(t_j) \subset S$, this $S(t_j)$ could be all or part of S , $\text{Card}(S(t_j)) \leq m$. The coverage relation between a sensor $s_i \in S$ and targets $t_j \in T$ is δ_{ij} could be represented by the binary matrix Δ below.

$$\Delta = (\delta_{ij}) = \begin{pmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \delta_{m1} & \delta_{m2} & \dots & \delta_{mn} \end{pmatrix}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if sensor } i \text{ can cover target } j \\ 0 & \text{otherwise} \end{cases}$$

Each sensor s_i has a limited initial energy reserve and the energy consumption rate for every monitoring period k equal to $E_i(k)$. The network lifetime L is aimed to be maximized regarding all the environment restrictions and the energy constraints. This lifetime could be defined and represented as an optimization problem with the WSNs lifetime maximization as an objective function. The WSN is normally assumed as dead when the available sensors failed to satisfy the required coverage. Therefore, in the case of all sensors available in S can monitor all targets in T , $T(S) = T$, then, activating all sensors can guarantee one monitoring season till all sensors run out of energy “if the sensors have the same initial energy and energy consumption rates”. To prolong the WSNs lifetime, it is preferable to find a collection or subsets, C_u and C_v of sensors with $T(C_u) = T(C_v) = T$ and $C_u \cap C_v = \Phi$, that guarantee several monitoring seasons “as more subsets will result in more monitoring seasons”. Or, if there is a collection C of subsets of sensors, $C = \{C_1, C_2, \dots, C_q\}$ with $T(C_l) = T(C_{l+1}) = T$ regardless of $C_l \cap C_{l+1} = \Phi$, a sensor can be activated on different monitoring seasons for a part of its energy. Also, the energy utilization could be optimally scheduled to guarantee a greater sum of monitoring seasons and optimize the WSNs lifetime. The subsets of sensors could be considered as covers, according to the following definition.

Definition 1. Cover of targets

Given a finite set $S = \{s_1, s_2, \dots, s_m\}$ of m sensors and a finite set $T = \{t_1, t_2, \dots, t_n\}$

of n targets, the collection of elements of S denoted C_l is a cover for the subset of targets denoted $T(C_l)$, if it can sense all the targets of $T(C_l) \subseteq T$, where:

1. Each target t_j is covered by at least one sensor of S .
2. Each sensor s_i senses a subset $T(s_i) \subseteq T$ of targets.
3. For all $i \neq j, T(s_i) \cup T(s_j) = T(s_i, s_j)$ is a subset of targets sensed by the pair (s_i, s_j) .
4. A subset C_l should be considered as a cover if $T(C_l) = T$.

Example 1: Sensors and targets coverage relation

Let S be a set of sensors with $m = 10$, $S = \{s_0, s_1, s_2, \dots, s_9\}$ used to monitor a set T of targets with $n = 10$, $T = \{t_0, t_1, \dots, t_9\}$ as in figure 2.1. One can see that s_1, s_2 and s_3 can cover all targets as an example.

The deployment of sensors in two-dimensional area consists exactly is finding the pair (x_i, y_i) for all $i = 1, 2, \dots, m$ and the targets positions in represented by the pair (x_j, y_j) for all $j = 1, 2, \dots, n$. The sensors deployment and the targets positions could be available, the deployment data in table 2.2 is randomly generated for this example.

TABLE 2.2: The sensor deployment and the targets positions

Index	Sensor		Targets	
	x	y	x	y
1	2	2	3	3
2	7	4	6	8
3	8	8	6	1
4	7	8	4	5
5	8	2	5	5
6	5	2	3	7
7	4	6	7	6
8	8	4	9	5
9	2	8	5	8
10	3	4	7	2

According to the sensors deployment, a sensor can cover the targets that are located within its coverage range and the coverage relations will be known while

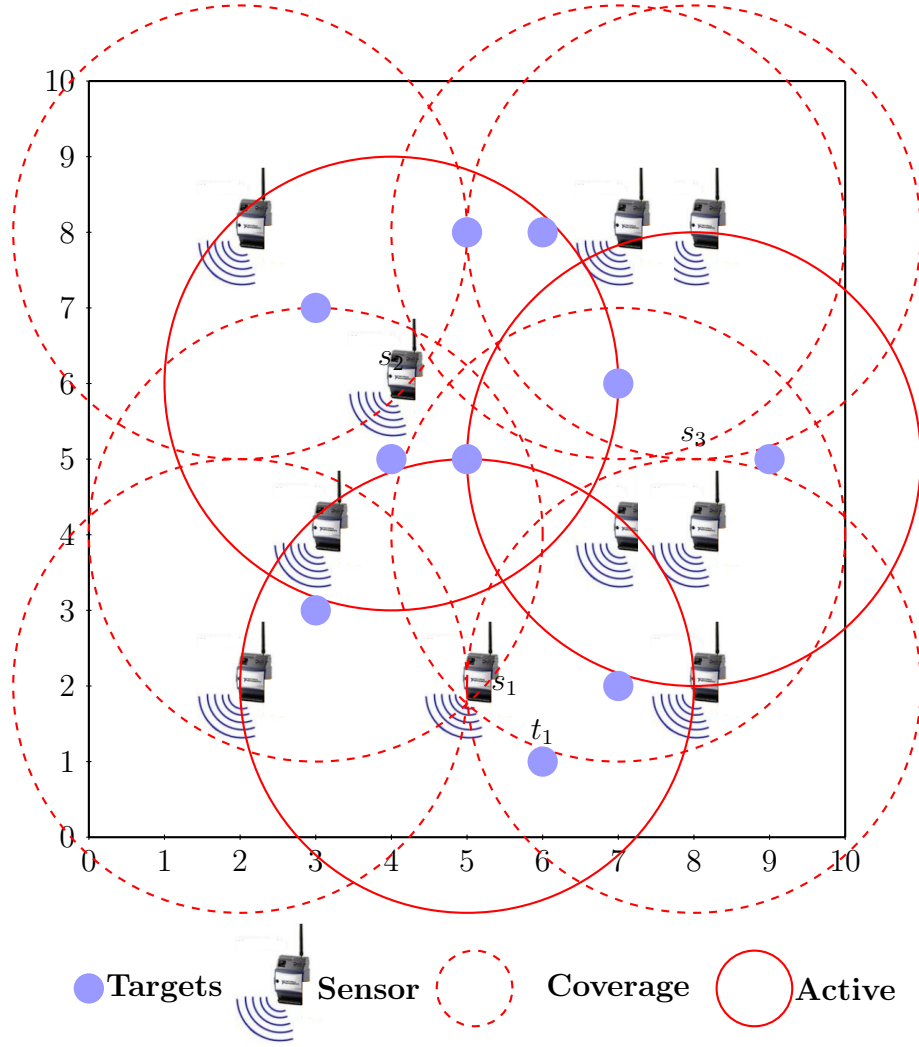


FIGURE 2.1: Sensors and targets relation.

the positions and sensing ranges are known for all sensors and targets. Therefore, the problem of target coverage is to find the subsets of sensors that can cover all targets to be scheduled to maximize the lifetime, considering the initial battery lifetime constraint to be satisfied regarding the energy consumption of all sensors.

Regarding the position of a target i , (x_j, y_j) and the distance d_{ij} between s_i and t_j , the coverage relation value δ_{ij} is equal to 1 if sensor s_i can cover the target t_j . That is the position of the target t_j is in the coverage range of sensor s_i or the distance d between the sensor s_i and the target t_j is less than or equal the coverage range r_i of sensor s_i , as:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$d_{ij} \leq r_i$$

and the relation matrix Δ for the sensors ($m = 10$) and targets ($n = 10$) is:

$$\Delta = (\delta_{ij}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The Δ matrix explains for every sensor $i \in S$ the targets that allocated in its coverage range $T(s_i)$ as the i^{th} row and the sensors that can cover every target $j \in T$ “ $S(t_j)$ ” as the j^{th} column.

2.3 Related works

Target coverage problem in WSNs is investigated as maximizing the lifetime of the network that must continuously monitor a set of targets. The sensors have limited battery power life and it is necessary to efficiently utilize the energy while monitoring all targets for maximum duration. This section mainly introduces the techniques used for sensors activity scheduling and optimization in a sensor network lifetime. There are several methods used to solve this problem. We organized these methods, as described in the following subsections, into categories namely: DSC based methods, exact methods and heuristics methods.

2.3.1 Disjoint Set Cover based methods

The problem of WSNs lifetime maximization is widely solved by transformation to a DSC maximization problem as in [10], [79], [80], [81], [82] and [83]. The DSC maximization problem is defined as it follows:

Considering a set $S = \{s_1, s_2, \dots, s_m\}$ of sensors used to monitor a set $T = \{t_1, t_2, \dots, t_n\}$ of targets. The objective function to be optimized is the maximal number of disjoint set covers as a representation of the network lifespan under the following restrictions:

1. A sensor can be included into one set cover at most.
2. Sensors in each disjoint set cover should be able to monitor all the targets.

Definition 1. Cover

Given a collection $C = \{C_1, C_2, \dots, C_q\}$ of subsets of a finite set S of sensors, find the maximum number of disjoint covers for set T of targets. Every cover C_i is a subset of S , $S(C_i) \subseteq S$, such that every element of T belongs to at least one sensor of $S(C_i)$, and for any two covers C_i and C_j , $C_i \cap C_j = \emptyset$ [79].

Definition 2. Disjoint Set Covers DSC

Given a collection C of subsets of S , a disjoint set cover $C_l \subseteq C$ is an element of C such that all elements of S belong to one and only one cover C_l ($l = 1, \dots, q \leq m$), i.e. for all $l \neq h$, $C_l \cap C_h = \emptyset$ [79].

If a number of DSCs is found, the expected scenario of DCSs utilization will be as in figure 2.2 below. From a set of sensors deployed to monitor a set of targets in figure 2.2.a, a subset of four sensors is activated to monitor all targets while the other sensors are in sleep mode as in figure 2.2.b. Figure 2.2.c explains that after the first subset is dead another subset of five sensors is activated for another monitoring season.

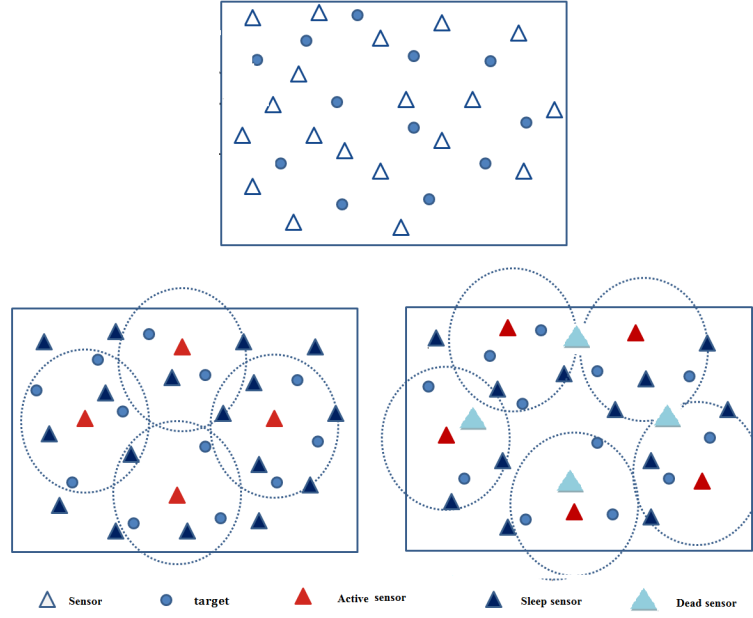


FIGURE 2.2: Sensors selection in covers.

Lemma 1. Solving the WSNs lifetime optimization problem as DSC problem is finding the maximum number of DSC for S .

Proof. From Definition 1, $T(C_u) = T(C_v) = T$, and from to definition 2, $S(C_u) \cap S(C_v) = \Phi$, $S(C_u)$ and $S(C_v)$ are partitions of S with $v \neq u$. Assuming that all sensors are identical, it is then obvious that the collection of all the sensors can sense all the targets on only one time period while consuming the energy of all the sensors. Assume now that there is a partition of S into two covers C_1 and C_2 such that $T(C_1) = T(C_2) = T$ with $T(C_1) \cap T(C_2) = \emptyset$, then the covers C_1 and C_2 can be scheduled to sense all the targets during two periods of time. Thus by increasing the number of disjoint set covers will correspondingly increase the number of period of operation of disjoint subsets of sensors. ■

Example 2: DSC illustration

Given the set of sensors and targets in example 1, let us try finding the maximal number of DSCs and the including relation IR matrix between covers and sensors. From the figure 2.1, the target t_1 is in the coverage area of only two sensors. Therefore, the maximum possible number of DSCs is two and the possible including

relation IR could be:

$$IR = (ir_{li}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$ir_{li} = \begin{cases} 1 & \text{if cover } l \text{ includes sensor } i \\ 0 & \text{otherwise} \end{cases}$$

Through all the methods used to solve the WSN lifetime optimization problem formulated as DSC problem, there is a valid number of sensors that have never been included in a cover or never used. Therefore, a valid amount of the WSN energy is not used. So, trying to implement all or part of this amount can extend the WSN lifetime.

The DSC problem has been proven in [79] NP-Complete, on the basis of partition properties in set theory. Through all these methods, a valid number of sensors is not included. In [10], a period iteration and sequential assignment heuristics were developed to maximize the network lifetime by using the maximum DSC, based on the sensors deployment and the sink node mobility. They have formulated the WSN lifetime maximization into a mathematical model considering the optimal sensor deployment, sensors activation scheduling, data routing and sink node mobility decisions. Also, they have provided two heuristics for the solution of this integrated formulation of the problem.

The authors in [79] mentioned that the number of the disjoint set covers increases when the number of sensors and the sensing range increase. The genetic algorithm is used in [83] to find the maximum number of covers for DSC problems using critical targets to find an upper bound. The sensors were randomly distributed into groups as candidate covers. Genetic algorithm with an operator called reconfiguration operator, considering the fitness and objective functions, was used to find better solutions and enhance WSNs lifetime [84]. Binary integer programming formulation and heuristics were also used in [80] to compute the maximum number of disjoint set covers for maximizing the WSNs lifetime. In [81], the targets system is

an indoor area divided into fields of points according to a finite resolution, where each sensor can cover one or more fields. The authors provided their heuristic for this problem to create the k-covers set. The authors in [79] transformed the problem of disjoint set covers maximization into maximum-flow problem (MFP) for maximizing the number of DSCs.

Example 3: the DSCs and NDSCs scheduling

Let $T = \{t_1, t_2, t_3, t_4\}$ and $S = \{s_1, s_2, s_3\}$, where $T(s_1) = \{t_1, t_2, t_4\}$, $T(s_2) = \{t_2, t_3, t_4\}$ and $T(s_3) = \{t_1, t_3, t_4\}$. As in figure 2.3, there are three possible covers.

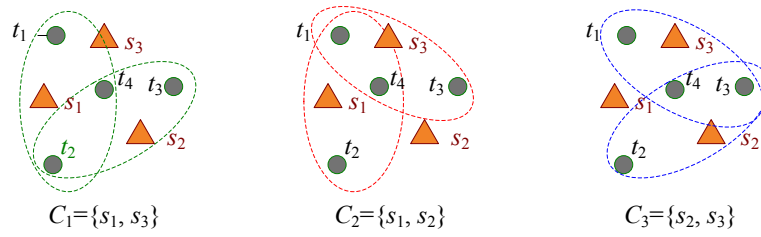


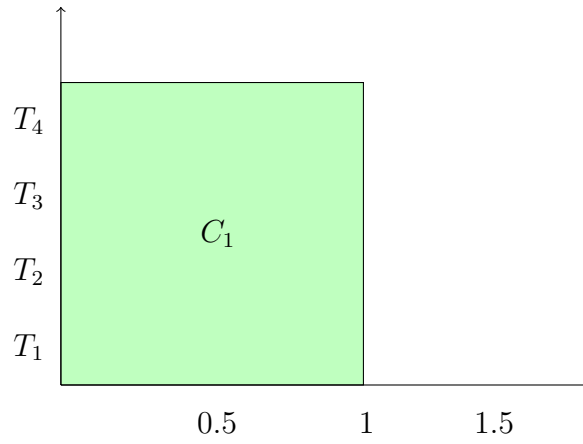
FIGURE 2.3: Sensors and possible covers.

It is clear that the maximum number of disjoint set covers is 1 “either C_1, C_2 or C_3 ” and therefore the network lifetime is 1 monitoring season with one sensor not included as in figure 2.4.a. While, if the sensors operated regardless of DSC as follows: $C_1 = \{s_1, s_3\}$ for $T(C_1) = 0.5$ unit of time, $C_2 = \{s_1, s_2\}$ for $T(C_2) = 0.5$ unit of time, and $C_3 = \{s_2, s_3\}$ for $T(C_3) = 0.5$ unit of time, the lifetime is 1.5 time unit as in figure 2.4.b [85].

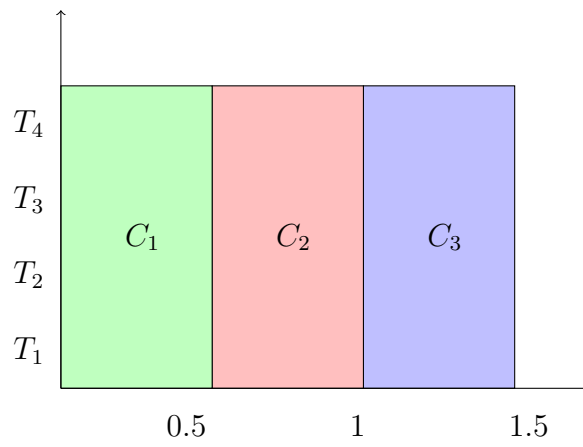
Therefore, the schedule of sensors can be formulated as: $S = \{s_1, s_2, s_3\}$ and $T = \{t_1, t_2, t_3, t_4\}$, where $T(s_1) = \{t_1, t_2, t_4\}$, $T(s_2) = \{t_2, t_3, t_4\}$ and $T(s_3) = \{t_1, t_3, t_4\}$. The maximum possible covers number $q = 3$ from figure 2.3. These covers can be scheduled as explained in figure 2.4.b. So, the sum of $t(C_l)$, $l = 1, 2, 3$, is the objective function, considering the sum of energy consumption $E_i(C_l)$, where $x_{il}=1$ if sensor i is included and the energy of each sensor i must not exceeds E_i . Thus the problem is formulated as follows:

Maximize

$$L = \sum_{l=1}^3 t(C_l) \quad (2.1)$$



a.DSCscheduling



a.NDSCscheduling

FIGURE 2.4: Covers scheduling.

subject to :

$$\sum_{l=1}^3 x_{il} E_i(t(C_l)) \leq E_i \text{ for } l = 1, 2, \text{ and } 3 \quad (2.2)$$

$$E_1(t(C_1)) + E_1(t(C_2)) \leq E_1$$

$$E_2(t(C_1)) + E_2(t(C_3)) \leq E_2$$

$$E_3(t(C_2)) + E_3(t(C_3)) \leq E_3$$

When the schedule satisfies this constraint, then the WSN lifetime is:

$$L = \sum_{l=1}^3 t(C_l) = t(C_1) + t(C_2) + t(C_3) \quad (2.3)$$

The optimal solution is

$$L = 1.5E \quad \text{with} \quad t(C_1) = t(C_2) = t(C_3) = 0.5E.$$

When the DSCs are used, it is clear that the maximum number of DSCs (and therefore network lifetime) is E with one sensor never included.

Example 1 gives an indicator that the lifetime can be extended better than using DSC if there is a possibility for a sensor or more to be used for a part of its lifetime while consuming a part of its energy in a cover, then the remaining part of its lifetime with the remaining part of its energy on another cover, as non disjoint covers or non-disjoint set covers (NDSC).

The proposed methods in this work aims to consider all possible covers to find out the optimal scheduling solution with possibility for sensors to be included in one or more covers.

2.3.2 Exact methods

Exact methods guarantee that the optimal solution could be found if a sufficient time is given for the algorithm. As stated earlier, a simple calculation explains that even if the exact methods used are based on more efficient techniques, the worst case of running time for NP-Hard problems are still going to be high [86]. In [87], the problem of maximizing the sensors network lifetime is formulated as a linear programming model, considering the energy cost for data sensing, receiving and transmitting. The linear programming technique is used to compute the maximal lifetime of the surveillance system and a workload matrix. Then, the workload matrix decomposed into a sequence of schedule matrices of monitoring seasons that can achieve the maximal lifetime.

In [68], linear programming is used to figure out the operational status of each sensor node either active or asleep considering the network coverage and minimizing energy consumption by minimizing the active nodes at a time as an objective

function. The linear programming mathematical model is formulated as equation 2.4 and 2.5.

minimize

$$\sum_{i=1}^m X_i \quad (2.4)$$

subject to :

$$\sum_{i \in Q_u}^m X_i \geq 1 \quad \forall u = 1, 2, \dots, w; \quad (2.5)$$

$$X_i \in 0, 1$$

Considering that X_i refers to sensor status “active 1 or sleep 0”, Q_u is the set of subscripts of nodes which can cover the area and w is the number of available set covers.

The authors used to search in 2^n possible combinations to find the one with the lowest energy consumption for this NP-hard problem, where n is the number of sensors. They show that linear programming can find the fewest number of nodes in working condition and achieve maximum network coverage thereby effectively extending the lifecycle of the networks. The ILP is also used in [88] to formulate the maximum lifetime broadcasting problems in WSNs. The authors show that a tool like ILP, often regarded as over-theoretical and unrealistic, are indeed suitable frameworks to include the latest advances in energy consumption and communication models in WSNs. The branch and bound (B&B) method is used in [70] to optimize the WSNs lifetime via DSCs maximization. A corresponding rule or function is defined for each step the three main steps: selection, branching and bound.

The authors have proven the B&B used is better in both maximum life time and running execution time compared to a Greedy-MSC algorithm proposed in [89]. Many researchers prefer supporting the exact method with heuristics to minimize the execution time and the computational effort required. Integer programming and greedy based heuristics are used in [90] to maximize the lifetime in a WSN with adjustable sensing ranges. Authors indicate that adjustable sensing ranges

have great impact on the network lifetime. Sensors with two coverage ranges and the possibility for a sensor to be included in more than one cover have been considered. In this case the sensors and targets coverage relation depends on the active sensing range as in the bipartite graph in figure 2.5, and also, the energy consumption rate. Mixed Integer is used in [90] for WSN lifetime maximization considering

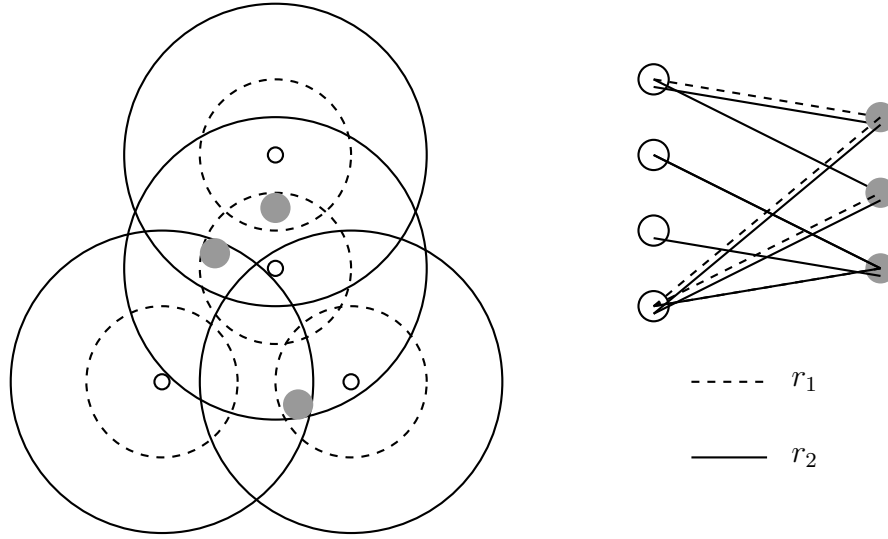


FIGURE 2.5: Two sensing ranges sensors and targets relations.

this two sensing ranges sensors and the base station mobility optimization

In [91], ILP approach is used alone and combined with GA as GA+ILP to address the problem of lifetime maximization of directional sensor networks. The authors found that the GA+ILP approach is about 4.74 times faster in average over all instances than the approach, which uses the ILP alone. The mathematical formulation of the problem of maximizing lifetime using a set of P covers as sum of the time τ_l assigned to each cover l considering the sensing directions d is illustrated in equations 2.6, 2.7 and 2.8.

maximize :

$$\sum_{l=1}^P \tau_l \quad (2.6)$$

subject to

$$\sum_{l=1}^P a_{il} \tau_l \leq E_i \quad \forall i \in \{1, 2, \dots, m\} \quad (2.7)$$

$$\sum_{l=1}^P a_{m+g+d(i-1)l} \tau_l \leq E_i \quad \forall i \in \{1, 2, \dots, m\}, \quad \forall g \in \{1, 2, \dots, d\} \quad (2.8)$$

$$\tau_l \geq 0, \quad \forall l \in \{1, 2, \dots, p\}$$

The a_{il} explains that the sensor i is a part of cover l considering the directions and $a_{n+g+d(i-1),l}$ is equal to 1 if and only if sensor i is part of C_l using its sector g for all $(i, g) \in \{1, \dots, m\} \times \{1, \dots, d\}$.

2.3.3 Heuristics and Meta-heuristics based methods

Heuristics are typical methods that can relatively quickly select a better candidate from feasible solutions with reasonable quality. There are no guarantees about the solution quality, though it can be arbitrarily bad. The heuristics are experimentally evaluated, and comments can be made about their quality of the heuristic solutions based on these experiments. Heuristics are typically used for solving real life problems because of their speed and their ability to handle large instances [86]. An optimal algorithm and heuristics are used in [92] for sensors activation/deactivation as depicted in figure 2.6 considering the coverage constraint and the target system health condition monitoring, to maximize the targets coverage time.

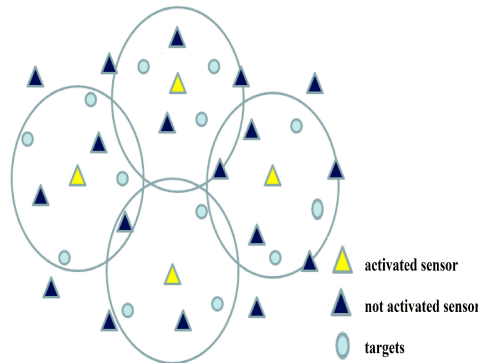


FIGURE 2.6: Sensors activation/deactivation.

A scheduling algorithm that allows to select just a minimum set of sensors to be activated considering the required coverage is developed in [93] to maximize the WSN lifetime considering the faults tolerance. Accordingly, the sensor node status is either asleep, work, dead or in test as in figure 2.7.

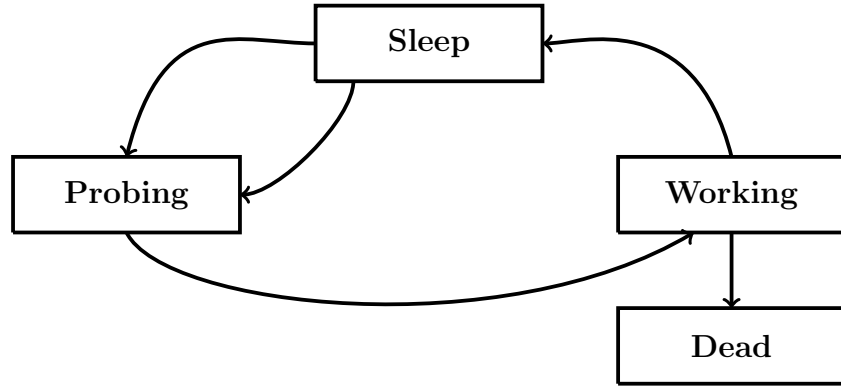


FIGURE 2.7: Sensors node status.

Authors in [94] consider the area coverage or maintaining, in which a full coverage of the monitoring area is required. The data request propagates through a broadcasting form the base stations to the sensing nodes and the data aggregation through which the sensing nodes transmit the information to the base stations. The author's effort is oriented to insure an energy efficient data propagation from nodes to the monitoring center. Authors in [80] developed a graph-based algorithm to be used for optimizing WSNs lifetime via solving the DSC maximization problem using specific mathematical formulation. They used the bipartite graph to find the optimal number of disjoint set covers through the following step:

- Step 1: Create the bipartite directed graph $G = (N, A)$: the node set is sensors and targets $N = S \cup T$ and the arc is the coverage relation δ_{ij} .
- Step 2: Find the target with a minimum number of sensors to cover (minimal $S(t_j)$) as a critical target t_j .
- Step 3: Based on, create a source node s_0 and create intermediate nodes s_{0i} for each sensor then create two sink nodes as an end points Y_1 and Y_2 .

A genetic algorithm combined with a genetic replacement algorithm is used in [95] to increase the number of active nodes considering reduces the rate of data loss and reduces the rate of energy consumption that increases the WSN lifetime. The GA-based algorithm is used based on binary-coded genes in a chromosome with a number of genes equal to the number of sensors.

The greedy algorithm is used in [96] for targets coverage scheduling in directional sensor networks (DSN). The authors have addressed the lifetime maximization via the maximum set covers for DSN then presented their targets coverage scheme, based on the following mathematical model used is:

maximize :

$$\sum_{l=1}^P \tau_l \quad (2.9)$$

subject to :

$$\sum_{k=1}^k \sum_{j=1}^d x_{igl} \tau_l \leq L_i \quad \forall s_i \in S \quad (2.10)$$

$$\sum_{g=1}^d x_{igl} \leq 1 \quad \forall s_i \in S, l = 1, 2, \dots, P \quad (2.11)$$

$$\sum_{D_{ig}} x_{igl} \geq 1 \quad \forall r_m \in R, l = 1, 2, \dots, P \quad (2.12)$$

The objective function in equation 2.9 illustrates the sum of the active time τ_l for all covers P which represents the covers in the DSN. Equation 2.10 is the sensors life constraint should be considered, it explains that the sum of the active times for all sensors “ $i = \{1, \dots, m\}$ ” included in all covers “ $l = \{1, \dots, P\}$ ” using all directions “ $g = \{1, \dots, d\}$ ” could not exceed the sensors life L_i .

Heuristic procedure is introduced in [97] for solving WSNs lifetime maximization problem using directional sensors as in figure 2.8. The authors have developed two greedy-based algorithms for solving the targets coverage problem in WSNs with adjustable sensing ranges. They used the critical targets as a bottleneck for

network lifetime to identify an upper bound on the maximum operation time of the network.

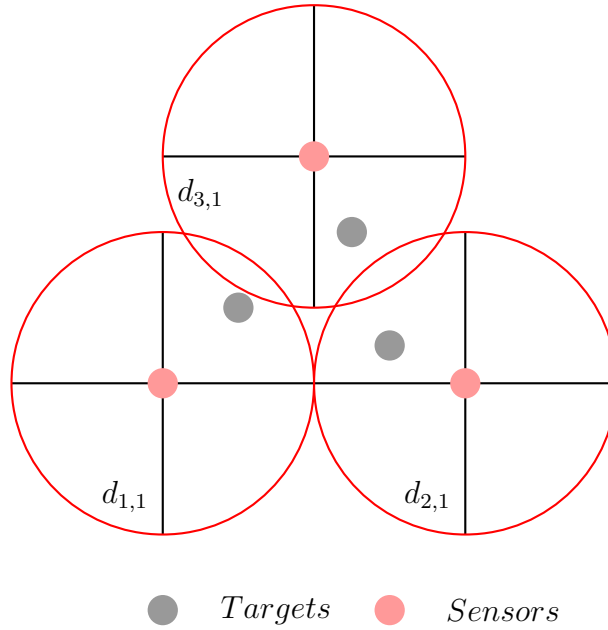


FIGURE 2.8: Directional sensors.

The figure illustrates a WSN with directional sensors $m = 3$ each sensor has a number of direction and monitoring sectors $d = g = 4$. Donato et al. [82] have compared greedy-based heuristics to an exact while considering an upper bound similar to that used in [97].

The column generation (CG) based heuristics were introduced by [98] and [99] for WSNs lifetime maximization. The (CG) is used in [98] for NDSCs WSNs lifetime maximization formulated in LP model and [99] have considered both coverage and connectivity.

2.4 Conclusion

Now, the problem is to find the optimal lifetime for the WSNs with a set of sensors $S = \{s_1, s_2, \dots, s_m\}$ used to monitor a set of targets $T = \{t_1, t_2, \dots, t_n\}$, given the followings hypotheses:

1. The targets positions (x_j, y_j) for $j = \{1, \dots, n\}$.

2. The sensors deployment data (x_i, y_i) for $i = \{1, \dots, m\}$.
3. Sensors initial amount of energy reserve E_i for all $i = \{1, \dots, m\}$.
4. Sensors energy consumption rate for the monitoring season k , $E_i(k)$ for all $i = \{1, \dots, m\}$.
5. Sensors limited sensing ranges d_i for all $i = \{1, \dots, m\}$.

Considering works on DSC for WSNs lifetime maximization introduced previously, the maximum lifetime obtained is not always the optimal energy utilization and lifetime for this network as depicted in example 3. that in addition to the obligation of using identical sensors in term of initial energy and energy consumption rates.

Solving this problem into NDSC investigated in this work removes the DSC constraint that gives the opportunity for a sensor to participate in more than one cover could perform better energy utilization and optimal lifetime for this network. This operating process offers: 1) the possibility to get longer lifetime using suitable algorithms, 2) the possibility for the algorithms to be implemented for not-identical sensor working in one group and 3) the possibility to reduce a sensor failure effect on the network lifetime and to maximize the network availability.

In NDSC, a sensor can participate in one or more covers. In this case, a sensor can spend part of its energy within a cover and another part within another one. Then, finding the optimal lifetime here produces two sub problems:

- **Finding the optimal number of NDSC:**

given the set S of sensors, set T of targets and the coverage relation matrix Δ , find the maximum number of NDSC q and the including relation matrix IR This problem of the maximum number of such cover sets, has been proved to be NP-complete as Zorbas et al. reported in [100].

- **Find scheduling manner that optimize the WSNs lifetime:**

based on the result from stage 1, given a collection $C = \{c_1, c_2, \dots, C_q\}$ of

NDSC, the initial energy E_i and the energy consumption for every monitoring season k for all sensors $E_i(k)$ find the optimal number of monitoring seasons that each cover in q should be scheduled $Y = \{y_1, y_2, \dots, y_q\}$ that optimize the WSNs lifetime L considering the coverage required and the energy consumption constraint satisfaction.

This work aims to find and investigate the optimal solution for the WSNs lifetime through the previous two subproblems. A binary representation and GA based heuristic are implemented to find the maximum possible number of NDSC. Then, for determining the optimize the WSNs lifetime, an integer linear programming model is formulated and GA-based algorithms are developed as will be described in the next chapter.

Chapter 3

Methods for Solving the Problem as Non-Disjoint Set Covers

3.1 Introduction

This chapter describes the methods that we developed to solve the problem of WSNs lifetime optimization, based on non-disjoint set covers approach. Different techniques were associated, for building an integrated method allowing to resolve this problem. The contribution of this work could be seen behind the formulation of the WSNs lifetime maximization as a scheduling problem, the mathematical modeling and the methods developed to solve this problem. Regarding the scheduling problem, we consider the environment as m non-identical parallel machine “sensors S ” with limited availability as a constraint and the objective is to maximize the number of jobs that executed on this environment. We represent a monitoring as a job over a time period k . That means the processing time for all jobs is equal to k , the period through which all targets should be monitored. Each job could be executed on one machine or more machines in parallel according to the subtasks; one sensor can monitor all targets or a subset of sensors is involved into the surveillance if one sensor could not. The subtasks st executed in parallel on all the machines must include all subtasks for the job in progress

“all targets T ”. Let us consider an environment with m non-identical machines, $m = 5$, used for jobs with 5 subtasks, $n = 5$ “ $Card(T)$ ”. Let’s define $st(m_i)$, the set of subtasks that could be scheduled for execution on machine i . For instance, $st(m_1) = \{st_1, st_3, st_4\}$, $st(m_2) = \{st_2, st_3, st_5\}$, $st(m_3) = \{st_1, st_4, st_5\}$, $st(m_4) = \{st_2, st_3, st_5\}$, and $st(m_5) = \{st_1, st_4\}$. This environment contains 5 machines with limited availability for each machine l_i for $i = 1, \dots, 5$ as in figure 3.1.a. The possibility of each subtask of all jobs to be executed on each of the machines could be presented as a bipartite graph $G(v, a)$ in figure 3.1.b. The vertex v is the machines and subtasks $m + n$ and the arches represent the possibility of execution relation. Considering that the processing time of each job is k , the jobs could be scheduled as in 3.1.c.

The objective function is the number of jobs that could be executed in this environment and must be maximized. As constraints for each job, all subtask should be performed in parallel as in equation 3.1.

$$\bigcup_{i=1}^m st(m_i) = J_j, \quad \forall j = 1 \text{ to } max \quad (3.1)$$

For all machines, the processing time for assigned jobs should not exceed the limited availability as in equation 3.2.

$$\sum_{j=1}^{max} v_{ij}k \leq L_i, \quad \forall i = 1, \dots, m \quad (3.2)$$

v_{ij} is a logical variable that explains the participation of machine i in executing job j . Thus, $v_{ij} = 1$ if participate or 0 if not. Regarding j_1 in figure 3.1.a, j_1 is executed on m_1 and m_2 , the first constraint from equation 3.1 become $st(m_1) + st(m_2) = T$ which should be satisfied for all jobs. In this phase, one must search for subsets of machines “sensors” that could handle a job as non-disjoint sets. The second constraint in equation 3.2 and the objective function describe the mathematical formulation of the optimal utilization of limited resources, which we have formulated as an integer linear programming mathematical model (ILP). Then, different approaches to the solution are investigated to reach for the optimal solution, by

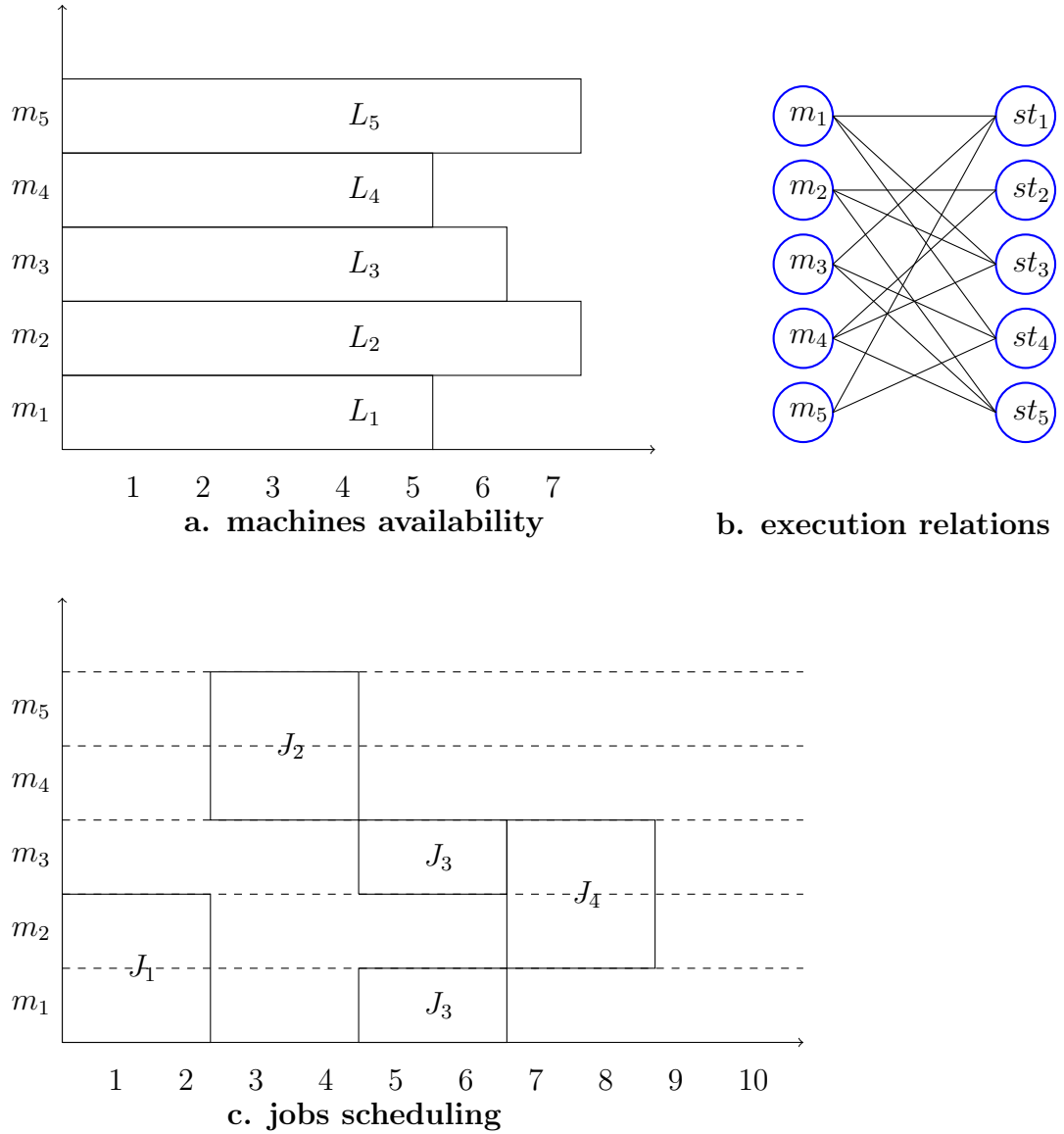


FIGURE 3.1: a schduling formulation of WSNs lifetime optimization

using both an exact method and heuristics. The problem is then solved in two stages: 1) finding the maximum number of NDSCs; and, 2) finding the optimal scheduling approach that maximizes the WSNs lifetime while considering its limited initial energy reserve. For the first stage, we have developed two methods: 1) the binary representation based method and 2) the GA-based method. Then, for the second stage, we have developed the mathematical model considering the number of non-disjoint set covers (NDSCs) generated using the first stage and a number of sensors with a limited initial energy. In this mathematical model, the objective function, to be maximized, is represented by the summation of the

number of monitoring seasons on that all covers are scheduled, considering that the energy consumption for all sensors included in all covers cannot exceed the initial energy stored in these sensors as a constraint. Then, to find the optimal solution we investigated two methods: 1) the integer linear programming algorithm to seek an exact solution, using the mathematical model of the problem; and, 2) the GA-based heuristics, used to reach the optimal solution in a reasonable time. Different GA-based heuristics are investigated, compared to different crossover and mutation operators. Also, a GA based on the DSC from the literature is described in this chapter and used for the evaluation of the optimal solution obtained using our methods. Figure 3.2 explains the comprehensive vision of the methods implemented. These methods will be described with more details in the next subsections.

3.2 Non-Disjoint Set Covers finding strategies

In NDSC, a sensor can participate in one or more covers. In this case, a sensor can spend part of its energy within a cover and another part within another one. Then, finding the optimal lifetime amounts to find the optimal number of NDSC and to optimize the lifetime by scheduling by utilizing of these covers. The problem of finding the maximum number of such cover sets has been proved to be NP-complete as reported in [100]. Indeed, authors proposed a centralized heuristic algorithm that efficiently generates cover sets, each of these covers is capable of covering all the targets.

3.2.1 The NDSC contribution to the DSC

In DSC, when a sensor i is classified to a cover C , this cover will be activated once until all the included sensors run out of energy “if the sensors were initially identically loaded”. Therefore, the total lifetime of this network is the sum of the lifetime of all DSCs and finding the optimal lifetime amount to finding the optimal

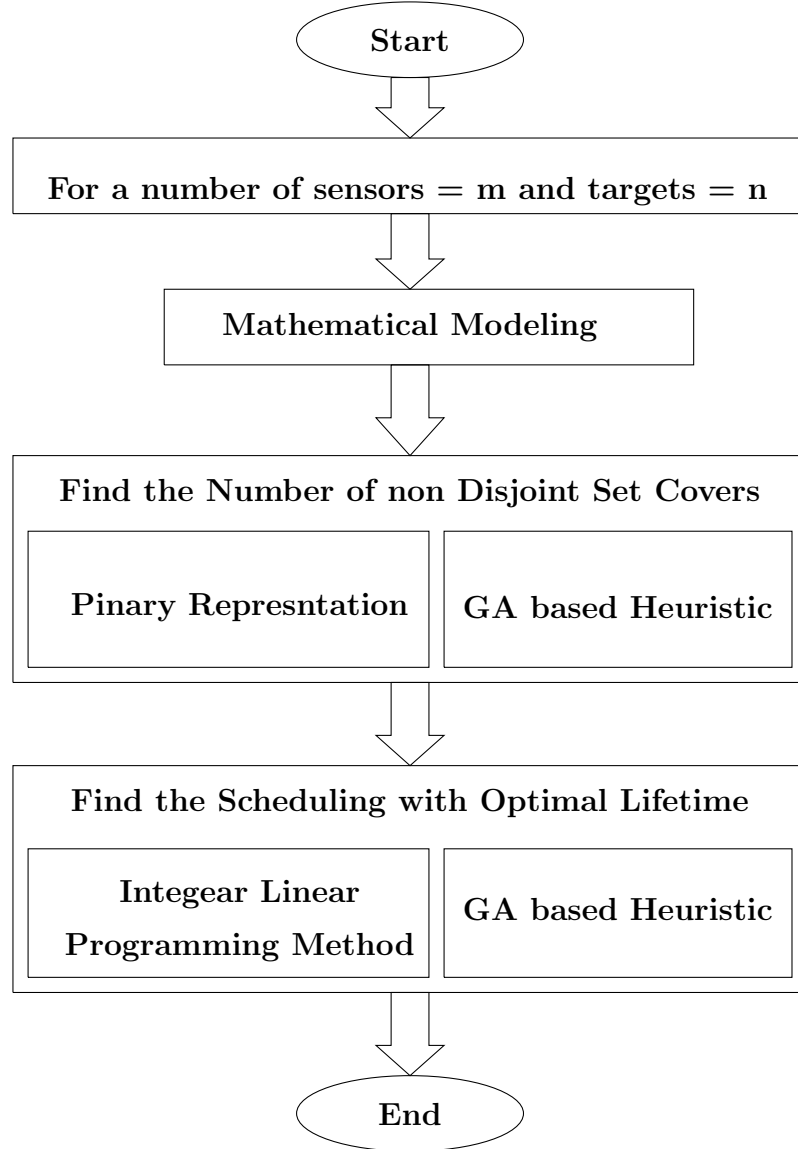


FIGURE 3.2: The method overview

number of DSCs regardless of the schedule in which these covers will be used. This is completely different from NDSC, where a sensor may enter one or more covers. An efficient coverage algorithm that can produce both disjoint cover sets, i.e. cover sets with no common sensor node, as well as non-disjoint cover sets were proposed in [101]. Moreover, the authors in [102] suggested their method that aims to reach the optimal lifetime for WSN organized as DSC. The main disadvantage of using DSC for WSNs lifetime maximization is that the maximum lifetime obtained based on DSC is not always the optimal lifetime for this network. We recall the example in [85] to explain this. Let $T = \{t_1, t_2, t_3, t_4\}$ and $S = \{s_1, s_2, s_3\}$, where $T(s_1) = \{t_1, t_2, t_4\}$, $T(s_2) = \{t_2, t_3, t_4\}$ and $T(s_3) = \{t_1, t_3, t_4\}$. Clearly, the

maximum number of disjoint set covers (and therefore network lifetime) is 1 unit of time with one sensor not included, while the sensors operated regardless of DSC as it follows: s_1 ; s_2 for 0.5 unit of time, s_2 ; s_3 for 0.5 unit of time, and s_1 ; s_3 for 0.5 unit of time, the lifetime is 1.5 time unit. Besides, in case of non-identically loaded sensors, the lifetime of the cover ends when a sensor expires, which means it depends on the sensor with the minimal initial energy or the sensor with max energy consumption rate.

For a set S of sensors used to monitor a set T of targets, the maximum lifetime coverage problem (MLCP) is defined as it follows: to find the optimal scheduling of covers to be activated and that maximizes the network lifetime, considering that all targets are continually covered by at least one sensor in the active cover and that the energy consumed by each sensor could not exceed its initial energy reserve. It is clear that removing the DSC constraint gives the opportunity for a sensor to participate in more than one cover. This operating process offers 1) the possibility to get longer lifetime using suitable algorithms 2) the possibility for the algorithms to be implemented for non-identical sensors working in one group 3) the possibility to reduce a sensor failure effect on the network lifetime and to maximize the network availability as mentioned before.

3.2.2 The binary representation based method

This method is used for sorting the deployed network to find all possible covers. For every sensor $s_i \in S$ and target $t_j \in T$, there is a pair (x_i, y_i) and (x_j, y_j) coordinates that should be known for sensors and targets location. Therefore, t_j is covered by s_i if the distance between them is less than or equal to the coverage range d_i of s_i . For a set S of sensors distributed for monitoring a set T of targets, each sensor s_i can cover a set $T(s_i) \in T$ of targets. Here, one aims to find q cover sets from the deployed sensors set S . These q cover sets could be found through the following steps.

1. Create the individual cover relations matrix Δ :

Based on the known sensors deployment data, targets positions and the

sensing range, the coverage relation between n sensors and m targets could be represented in binary matrix Δ as:

$$\Delta = (\delta_{ij}) = \begin{pmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \delta_{m1} & \delta_{m2} & \dots & \delta_{mn} \end{pmatrix}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if sensor } i \text{ can cover target } j \\ 0 & \text{otherwise} \end{cases}$$

δ_{ij} is equal to 1 if the distance d_{ij} between sensor s_i and target t_j is less than or equal the coverage d_i range of sensor s_i . In this matrix, a row i represents the set of targets $T(s_i)$ covered by sensor s_i and each column j represents the set of sensors $S(T_j)$ that can cover target t_j .

2. Create the set cover matrix V , which models the relations between sensors and covers for m sensors and the targeted q covers:

To find the q covers, let us find all the feasible candidate covers or search space. This search space includes all possible permutations of the m sensors included into covers, starting from one sensor included representing a cover, to all sensors included in one cover, which corresponds to $2^m - 1$ candidates. The candidate covers could be represented in temporary binary matrix W as:

$$W = (w_{li}) = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ w_{(2^m-1)1} & w_{(2^m-1)2} & \dots & w_{(2^m-1)m} \end{pmatrix}$$

$$v_{li} = \begin{cases} 1 & \text{if sensor } i \text{ is included in cover } l \\ 0 & \text{otherwise} \end{cases}$$

Assuming that q is the number of possible covers $q \leq 2^m - 1$ then, each candidate will be considered as a cover and moved into the final including relation matrix V if the sensors included in this cover are capable of monitoring all targets in T . The final including relation matrix V between the q covers and the m sensors could be represented as:

$$V = (v_{li}) = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_{q1} & v_{q2} & \dots & v_{qm} \end{pmatrix}$$

$$v_{li} = \begin{cases} 1 & \text{if sensor } i \text{ is included in cover } l \\ 0 & \text{otherwise} \end{cases}$$

Now, finding a cover set with a minimum number of sensors in V is possible. From the previous step, each row covers completely the targets. The summation of each row in V gives the number of sensors in this cover. So, finding the cover with minimum sensors amounts to finding the row with minimum sum.

3.2.3 Genetic Algorithm based method

For m sensors used to monitor n targets, there are $2^m - 1$ possibilities of sensors clustering into covers as in W matrix. Referring to the coverage relation matrix defined in the Δ matrix, a row i denoted $T(s_i)$ represents the targets covered by sensor s_i and each column j represents the set of sensors denoted $S(T_j)$ that can cover target j . Then, the including set cover matrix V should be created to express the including relations between the sensors m and the targeted covers q as in V matrix.

According to the binary representation of sensors into covers, from one sensor included up to all sensors included, without empty cover, $(2^m - 1)$ is the total number of possible covers to be evaluated. This method is applicable for small number of sensors (for 5, 10 and 20, the solutions in $(2^5 - 1)$, $(2^{10} - 1)$ and $(2^{20} - 1 = 1048575)$ because of the exponential growth of the problem complexity. For such problems, heuristic or meta-heuristic algorithms are efficient for searching the optimal solution in reasonable time as it is well known.

For the optimal solution search to this problem, a GA-based heuristic was developed to 1) offer the network extension possibility, 2) get benefit from heuristics and its capability to find a near-optimal solution in reasonable time, 3) get the advantages of GA as an efficient technique for solving such kind of problems. This method could be integrated into an efficient scheduling algorithm or strategy to reach a satisfactory near-optimal lifetime for WSN clustered into NDSC. We propose a genetic algorithm for nun-DSC problem as follows:

1. **Coding and initialization:** For a set S of sensors used to monitor a set T of targets, the input of this algorithm is the coverage relation matrix which synthesizes the targets $T(s_i)$ from T that are covered by each sensor s_i from S . The output of this algorithm is the maximum number q of covers and the including relation matrix that explain the sensors from S included in each cover C_l from q that. This considering that for all C_u and $C_v \in C$ with $C_u \neq C_v$, it is possible for $c_u \cap c_v$ to be ϕ or more. For GA, we generate chromosomes with a number of genes equal to the number of sensors as a candidate cover. Each gene in the chromosome corresponds to a sensor from S , and its value expresses as its membership of this cover. The gene value is equal to 1 if the sensor is included in this candidate cover and 0 otherwise. The initial set of candidate covers is randomly generated using specific population size.
2. **Fitness:** The number of targets that a candidate can cover is considered as fitness value of this candidate. The candidate will be with two as a cover if the fitness value of this candidate is equal to T as illustrated in equation

3.3.

$$\bigcup_{i=1}^m v_i T(s_i) = T \quad (3.3)$$

3. **Selection:** The candidates with greater fitness value T will be selected as covers to the cover set. Then, all the population members are considered to be involved in the next stages of the GA for more covers to be added to the covers set in initial including matrix V .
4. **Crossover** and new candidate covers generations: Crossover is generally used to generate the new population from selected parents. In our approach, we used it to generate the next generation for all population taking every two members as parents.
5. **Mutation:** The mutation aims at enhancing the inherited value of the solutions. While the gene value is either 0 or 1, a randomly selected gene can be inserted. But, one can say it is better to a gene with value 0 to be 1 to increase the number of sensors in the candidate, which increases the possibility to get cover. The strategy used is randomly select a gene and set it to 1 to specify inclusion of the selected sensor into this cover.

Repeat step 2 and 5 for a specific number of iterations and add the new fit candidates in each iteration to the including matrix V with duplicate avoidance to generate the final version of the including matrix V as a result.

Recalling back the DSC, when the covers are found, there are no additional efforts required for scheduling and finding the optimal energy distribution through the covers because the DSC guarantees that there are no shared sensors. Now, removing the DSC constraint, a sensor can join more than one cover. Thus, there is another subproblem to be solved; that is how much of time or monitoring seasons a sensor from S should spend with each cover to achieve the optimal WSNs lifetime. The following subsections mathematically formulate this problem and propose exact and heuristic methods to solve it.

Example 1: The GA based method illustration

Let $T = \{t_1, t_2, t_3, t_4, t_5\}$ and $S = \{s_1, s_2, s_3, s_4, s_5\}$, where $T(s_1) = \{t_1, t_3, t_5\}$, $T(s_2) = \{t_2, t_3, t_4\}$, $T(s_3) = \{t_1, t_4, t_5\}$, $T(s_4) = \{t_2, t_4, t_5\}$ and $T(s_5) = \{t_1, t_2, t_3\}$ with $Card(T) = n = 5$ and $Card(S) = m = 5$.

The voverage relation matrix is:

$$CR = (cr_{ij}) = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \quad (3.4)$$

The approach we proposed based on GA could be used as follows:

1. Coding and Initial population:

A chromosome with 5 genes should be used for randomly generated initial population with known size “assume 10” as in the *In_pop* matrix below.

$$In_pop = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad (3.5)$$

2. Fitness and selection

From the initial population in the *In_pop* matrix, select the candidate with the fitness value equal to T using the following function.

$$\bigcup_{i=1}^5 v_i T(s_i) = T \quad (3.6)$$

According to equation 3.6 some of the candidates from *In.pop* in equation 3.5 could be selected to the first version of the including relation matrix V

$$V = (v_{li}) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad (3.7)$$

3. Crossover and mutation

Consider every two of the population as parents and apply a one-point crossover and one-point mutation to update the population.

Repeat step 2 and 3 for a given number of iterations and add the new fit candidates in each iteration to the including matrix V with duplicate avoidance to generate the final version of the including matrix V as a result.

3.3 Scheduling and optimization strategies

In all the methods developed to solve the WSN lifetime optimization problem formulated as DSC problem, there is a valid number of sensors that have never been included in a cover or never used. Therefore, a valid amount of the WSN energy is not consumed. So, trying to consume all or part of this amount of energy via NDSCs can extend the WSN lifetime. We introduced in the previous subsection the methods for generating a number q of NDSCs with corresponding the including relation matrix V . The present subsection aims to describe the methods used to search for the suitable period of time that each of the q covers should be scheduled, so as to optimize the WSNs lifetime, considering the limited initial energy reserved for all sensors. The method suggested a small period of time k for the monitoring season. Thus, each cover C_l for $l \in \{1, 2, \dots, q\}$ can be scheduled for an number of monitoring seasons y_l . Then, the lifetime to be optimized is the sum

of the monitoring seasons y_l scheduled for all cover in q . The sum is aimed to be maximized considering that the energy consumption through all sensors included in all covers must not exceed the initial energy of these sensors. The following subsections explain the mathematical model, the exact method based on the ILP and the heuristics based on GA which we developed to find the optimal solution for this problem.

3.3.1 The mathematical model

For the mathematical modeling of the WSN lifetime, let $S = \{s_1, s_2, \dots, s_m\}$ be the set of sensors used to monitor a set of targets $T = \{t_1, t_2, \dots, t_m\}$. Each sensor $s_i \in S$ can cover a set of targets $T(s_i) \leq T$ and loaded with a given amount of energy E_i . In the previous subsection we showed the methods used to generate a collection of covers $C = \{C_1, C_2, \dots, C_q\}$ with the corresponding including relation matrix V . A row from V incorporates the sensors from S that each cover C_l includes: the sensor s_i is included in cover C_l if $v_{li} = 1$. Each cover $C_l \in C$, and so the included sensors, can be scheduled for a number of monitoring seasons y_l . Therefore, the energy consumption of a sensor s_i included in cover c_l that is scheduled for a number of monitoring seasons y_l is $E_i(y_l)$ can be computed. On this base, the WSNs lifetime is determined by the summation of the scheduled monitoring y_l for all $l = \{1, 2, \dots, q\}$, the objective function to be maximized being the lifetime. The energy consumption for a sensor $s_i \in S$ that is included in one or more covers from C is the summation of all $E_i(y_l)$ where s_i is included " $v_{li} = 1$ ". The objective function and the energy constraint could be formulated as follows: For a set S of sensors ($S = \{s_1, s_2, \dots, s_m\}$) used to cover a set T of targets ($T = \{t_1, t_2, \dots, T_n\}$), the problem of maximizing the network lifetime amounts to find the optimal number of monitoring seasons y_l for the cover C_l for all $l \in \{1, 2, \dots, q\}$ that can cover all the targets in T .

The Objective Function

If cover C_l is scheduled to monitor the targets T for a number of period k of time units equal to y_l and the maximum possible number of covers is q , then the lifetime

maximization problem can be written as in equation 3.8.

$$\text{Maximize } k \times \sum_{l=1}^q y_l \quad (3.8)$$

Constraints

The total energy consumed by each sensor s_i cannot exceed its initial energy reserve E_i .

If the total energy consumed by a sensor s_i on the period of time k is $E_i(k)$ and sensor s_i can be included in every cover c_l of q "when $v_{li} = 1$ " which can be scheduled for a number of periods y_l , then the energy constraint can be written as in equation 3.9.

$$\sum_{l=1}^q v_{li} E_i(y_l) \leq E_i \quad i = 1, 2, \dots, m \quad (3.9)$$

with

$$v_{li} = \begin{cases} 1 & \text{if sensor } i \text{ is included in cover } l \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the optimal solution is the solution that can give the values of y_1, y_2, \dots, y_q with the maximum sum, while considering not exceeding the sensors initial energy E_i . We used the ILP and GA for exact and heuristic methods respectively to solve this problem as described in the next subsections. The example below helps to figure out the problem.

Example 2: the NDSC illustration

Let $T = \{t_1, t_2, t_3, t_4\}$ and $S = \{s_1, s_2, s_3\}$, where $T(s_1) = \{t_1, t_2, t_4\}$, $T(s_2) = \{t_2, t_3, t_4\}$ and $T(s_3) = \{t_1, t_3, t_4\}$. The coverage relation matrix Δ is:

$$\Delta = (\delta_{ij}) = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

To find the number of NDSCs q and the including relation matrix V , the binary representation or GA based method are used to search in the $2^3 - 1$ possible candidates in W as:

$$W = (w_{li}) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

The including relation matrix V found from the W is

$$V = (v_{li}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Therefore, the maximum possible number of covers $q = 3$. These covers can be scheduled as depicted in figure 3.3.

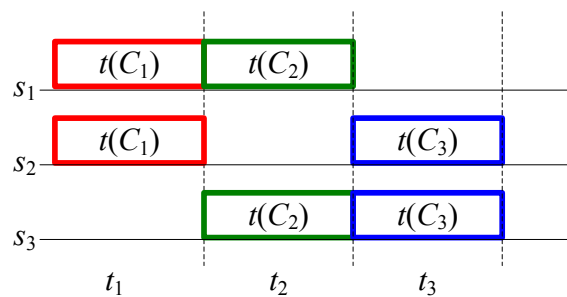


FIGURE 3.3: Covers scheduling.

So, the sum of $t(C_l)$ “ $y_l \times k$ ” is the objective function to be optimized, admitting that the sum of energy $E_i(C_l)$ consumed by the sensor s_i involved in C_l such that $v_{li} = 1$, is less than or equal to the initial energy E_i for each sensor i , as constraint. The objective function to be optimized and the energy consumption constraints

for this example become in equation 3.10 and 3.11

$$L = \sum_{l=1}^3 t(C_l) = \sum_{l=1}^3 k \times y_l = k \times \sum_{l=1}^3 y_l \quad (3.10)$$

subject to:

$$\sum_{l=1}^3 v_{li} E_i(t(C_l)) \leq E_i \text{ for } i = 1, 2, \text{ and } 3 \quad (3.11)$$

$$E_1(k \times y_1) + E_1(k \times y_2) \leq E_1$$

$$E_2(k \times y_1) + E_2(k \times y_3) \leq E_2$$

$$E_3(k \times y_2) + E_3(k \times y_3) \leq E_3$$

Or:

$$y_1 + y_2 \leq E_1/E_1(k)$$

$$y_1 + y_3 \leq E_2/E_2(k)$$

$$y_2 + y_3 \leq E_3/E_3(k)$$

When the DSCs based method is used, it is clear that the maximum number of DSCs (and therefore network lifetime) is $1E$ with one sensor never included.

Example 2 gives an indicator that the lifetime can be extended better with NDSCs than using DSC. The possibility for a sensor or more to be used for a part of its lifetime with a part of its energy in a cover, then the other part of its lifetime with the other part of its energy in another cover as non-disjoint set covers could prolong the lifetime. The proposed methods consider all possible covers to find out the optimal scheduling with the possibility for sensors to be included in one or more covers as depicted in the next subsections by using respectively ILP and GA.

3.3.2 Integer Linear Programming based method

Integer linear programming refers to the class of combinatorial constrained optimization problems with integer variables, where the objective function is a linear function and the constraints are linear inequalities [103]. Based on the previous subsections, there is a set of q covers generated for a WSN with a set S of sensors used to monitor a set T of targets that should be scheduled to maximize the lifetime of this WSN. As mentioned before, different methods are used in the literature to solve the lifetime maximization problem represented into DSCs maximization problem with various strategies such as GAs. We have designed a linear model for the lifetime as an objective function to be maximized and the energy consumption for each sensor as constraints not to be exceeded as in the previous subsection.

Let us consider a small period of time k as a monitoring season and the energy consumption of a sensor $s_i \in S$ activated for this season is $E_i(k)$. For a cover C_l scheduled to monitor the targets T for a number of monitoring seasons y_l with k units of time for each season and the maximum possible number of covers is q , then the lifetime can be calculated as illustrated in equation 3.12.

$$L = k \times \sum_{l=1}^q y_l \quad (3.12)$$

Considering that the energy consumed by a sensor s_i for a period of time k is $E_i(k)$ and every sensor $s_i \in S$ can be included in every cover $C_l \in C$, then the energy consumed by a sensor s_i on all periods over which cover l is scheduled is $E_i(y_l)$. Considering that the summation of energy consumption cannot exceed its initially reserved energy E_i then:

$$\sum_{l=1}^q E_i(k \times y_l) V_{li} \leq E_i \quad i = 1, 2, \dots, m \quad (3.13)$$

Given a WSN with m sensors and q possible covers, equation 3.13 comprises m constraints expressed in form of linear inequations with q unknowns ($q \leq m$).

Therefore, the optimization technique through linear programming can be used to find out the values of the unknowns that optimize the linear objective function L in equation 3.12.

When the initial energy of each sensor $s_i \in S$ is constant E_i and the energy consumed by each sensor s_i on a period of time k is constant $E_i(k)$, the maximum possible number of sensing periods for sensor s_i to be used is $E_i/E_i(k)$. Therefore, the cover $l \in \{1, 2, \dots, q\}$ that includes sensor s_i could not be used for more than this limit. Thus, for all cover $l \in \{1, 2, \dots, q\}$, the number of periods y_l on which cover C_l can be scheduled is bounded between 0 and $E_i/E_i(k)$ as illustrated in equation 3.14.

$$0 \leq y_l \leq E_i/E_i(k) \quad l = 1, 2, \dots, q \quad (3.14)$$

Equation 3.14 explains that y_l is zero or positive value $y_l \leq E_i/E_i(k)$.

All the decision variables in equations 3.12, 3.13 and 3.14 are integers and the equations are linear. Therefore, the ILP is implemented to find the exact solution for this problem.

Example 3: the ILP illustration

Let a WSN with 3 sensors and 5 targets, i.e. $S = \{s_1, s_2, s_3\}$ $T = \{t_1, t_2, t_3, t_4, t_5\}$, with sensor covers $T(s_1) = \{t_1, t_3, t_4\}$, $T(s_2) = \{t_2, t_3, t_4, t_5\}$ and $T(s_3) = \{t_1, t_2, t_5\}$ as in example 1. The maximum possible number of covers $q = 3$: $C_1 = \{s_1, s_2\}$, $C_2 = \{s_1, s_3\}$ and $C_3 = \{s_2, s_3\}$. Therefore the lifetime to be optimized can be represented as:

$$L = k * \sum_{l=1}^3 (y_l) = (y_1 + y_2 + y_3) * k \quad (3.15)$$

Considering the energy consumption constraint in equation (3.11) which can be represented with 3 linear inequalities of 3 unknowns.

$$\sum_{l=1}^3 E_i(y_l) V_{li} \leq E_i \quad i = 1, 2, 3 \quad (3.16)$$

$$E_1(y_1) + E_1(y_2) \leq E_1$$

$$E_2(y_1) + E_2(y_3) \leq E_2$$

$$E_3(y_2) + E_3(y_3) \leq E_3$$

For all cover $l \in \{1, 2, 3\}$, y_l should be bounded as:

$$0 \leq y_l \leq E_i/E_i(k) \quad l = 1, 2 \text{ and } 3$$

The optimal solution can be found out by solving this linear system (3.15) and (3.16).

Enhancement and simplification

When the number of covers increases, the number of unknowns increases. We used the following lemma 2 to reduce the number of covers. We ignored part of the covers to enhance and simplify the method without affecting its efficiency.

Lemma 2: For all covers C_u and $C_v \in C$ with $S(C_u) \neq S(C_v)$ and $S(C_u) \cap S(C_v) = S(C_u)$, it is better to use C_u and ignore C_v to achieve minimum energy consumption.

Proof: Considering the energy cost for monitoring the T targets for a period of time Δt , the energy consumption using cover C_u and C_v is $E_{S(C_u)}(\Delta t)$ and $E_{S(C_v)}(\Delta t)$ respectively, where $E_{S(C_u)} = E_i(k)$ for all $s_i \in S(C_u)$ and $E_{S(C_v)} = E_i(k)$ for all $s_i \in S(C_v)$. Therefore, $E_{S(C_v)}(\Delta t) > E_{S(C_u)}(\Delta t)$ while the sensors in $S(C_u)$ are a part of $S(C_v)$ but not all. ■

Example 4: the Enhancement and simplification method illustration

For $S = \{s_1, s_2, s_3, s_4, s_5\}$ and $T = \{t_1, t_2, t_3, t_4, t_5\}$ randomly deployed in 10×10 area, that is $m = 5$ and $n = 5$, the coverage relation matrix is:

$$CR = (cr_{li}) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

There are $2^5 - 1 = 31$ possible arrangements of sensors into covers distribution. Only 15 of them are covers as in the including relations matrix:

$$V^T = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

By applying the enhancement and simplification method, the cover relations matrix V becomes:

$$V = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

in which the number q of NDSCs is decreased from 15 to 4 using the enhancement and simplification method. The decrement could decrease the model complexity by decreasing the number of decision variables as illustrated by equation 3.12.

3.3.3 Genetic Algorithm based method

In general evolutionary algorithms vision, genetic programming began as a general model for an adaptive process but has since become effective optimization strategy [72]. By the end of the previous subsection, we obtain a number q of covers that should be scheduled to maximize the network lifetime. As mentioned earlier, different methods are used in the literature to solve the lifetime maximization problem such as DSC linear programming, with many strategies.

In this subsection, we introduce the GA-based scheduling method for WSNs lifetime optimization using NDSCs. The number q of NDSCs generated using the binary representation, or GA-based method, and the previous mathematical model are used. The WSNs lifetime to be optimized is represented as a summation of the number of monitoring seasons y_l scheduled for each cover $C_l \in C$ considering the energy consumption constraint. The value of y_l is bounded between *zero*

and $E_i/E_i(k)$. The exact solution for this problem is not reachable for greater instances because of the effort required in terms of execution time and computation resources. Normally, heuristics and meta-heuristics are used to obtain a near-optimal solution in reasonable time with the available computation resources.

A simple effective scheduling method using genetic algorithm is developed to search for the optimal values of y_l scheduled for all covers $C_l \in C$ that maximize the WSNs, considering the energy consumption constraint as follows:

1. *Coding*

The chromosome is represented as a vector of integer values of y_l between zero and $E_i/E_i(k)$ “the sensors initial energy E_i divided by the energy cost of sensor activation for a period of time k ($E_i(k)$ is the battery lifespan”. The number of genes, which is the vector length is equal to the number of covers q , and each gene represents the number of periods for each cover to be scheduled as in figure 3.4.

Periods for C_1	Periods for C_2	...	Periods for C_q
-------------------------------------	-------------------------------------	------------	-------------------------------------

FIGURE 3.4: Genetic algorithm chromosome creation.

2. *Initialization*

The initial population of a limited number of chromosomes is randomly generated. A gene represents the number of seasons scheduled for each cover. Therefore, the sum of genes in the chromosome represents the total number of seasons that all covers are scheduled, which is the WSNs lifetime in seasons. Thus, for each chromosome, the lifetime can be calculated by the sum of genes in the chromosome multiplied by the period's length of time k .

3. *Fitness*

The fitness has to ensure that all sensors in the candidate solution did not exceed the energy constraint. For a sensor $s_i \in S$, the sum of the energy consumption through all covers in a schedule should not exceed the initial

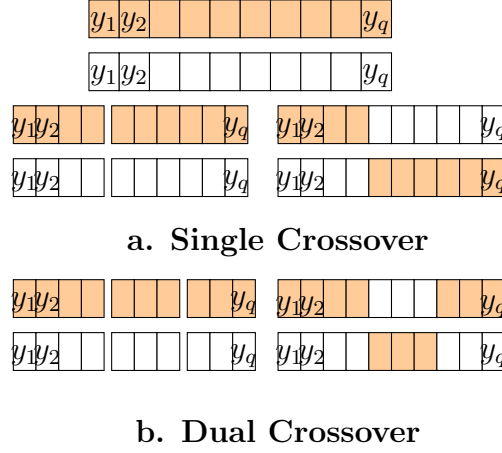


FIGURE 3.5: Genetic algorithm crossover strategies.

energy E_i . The fitness function will exclude the candidate schedule in the population that not meet this constraint from the next GA processes.

4. *Selection*

From the fit population members, select the best two candidate schedules with the maximum sum of genes (lifetime) as parents.

5. *Crossover*

A suitable crossover is applied to generate children for the next generation and update the population. Single and dual points crossovers are used as in figure 3.5.a and 3.5.b respectively. The crossover points are selected randomly as normally used in GAs. Different crossover operators are investigated as mentioned in the next paragraphs.

6. *Mutation*

A suitable mutation is applied to enhance the children and avoid the local optimal solution. One or more genes are selected randomly and updated. We consider that the enhancement increases the genes value, and thus, increases the sum of the chromosome. The mutation genes are selected randomly. We have investigated different increasing strategies and operators generate the new value of the selected gene.

7. *Children fitness*

Regarding the energy consumption in equation (3.11), the fitness function is

applied again for the children “new chromosomes” to make sure that no sensor exceeds its initial energy. The parents updated if new enhanced childrens have better sum of genes or lifetime.

New generations may be obtained by continuously repeating the above stages until reaching a specific upper bound, or the specified number of iterations.

3.3.3.1 Crossover operators

The GA crossover and mutation operators are presented in figure 3.6 with the possible permutations, considering the initialization, fitness, selection, mutation and crossovers.

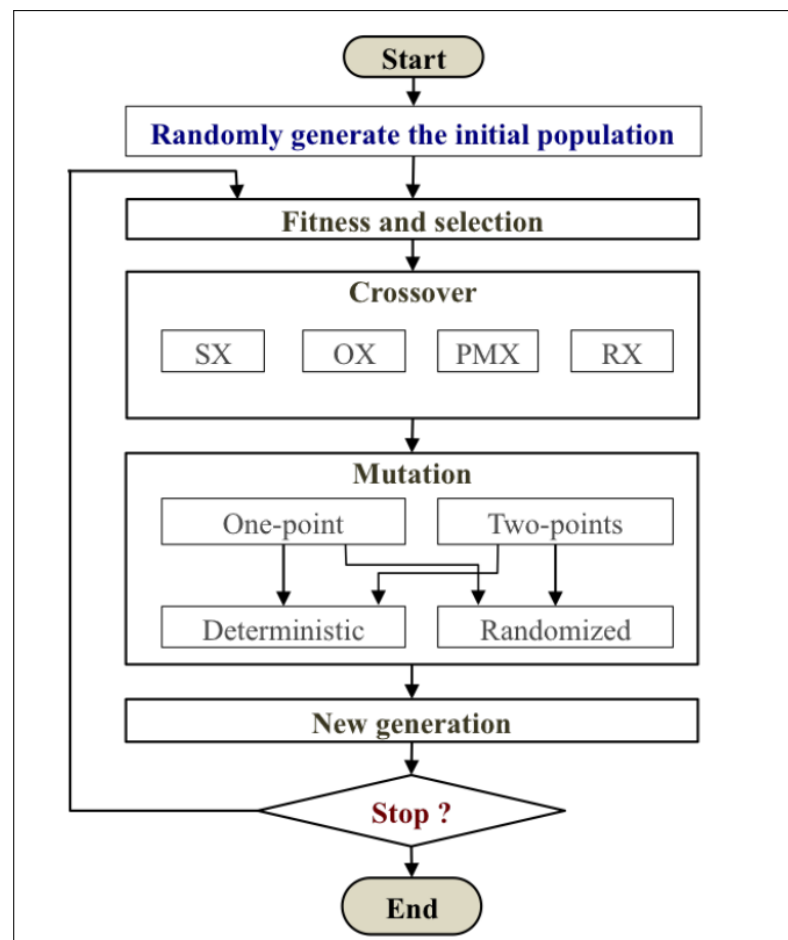


FIGURE 3.6: The Genetic algorithm Combinations

The simple crossover (SX), partially matched crossover (PMX), rotated crossover (RX) and order crossover (OX) described in the next paragraph are used.

- **Simple crossover (SX)**

For the SX, as the traditional one-point crossover, one crossing point is randomly selected and the parents exchange their parts to generate new children for the next GA processes.

- **Partially matched crossover (PMX)**

For both PMX and OX, when two strings $A = y_0, y_1, y_2, \dots, y_{q-1}$ and $B = y_0, y_1, y_2, \dots, y_{q-1}$ chromosomes are also selected, two random crossover points c_1 and c_2 are selected. The chromosome has now three parts: y_0 to y_{c_1} , y_{c_1} to y_{c_2} and y_{c_2} to y_{q-1} . Then, in PMX, crossover strategy is:

- the parents crossing: the part y_{c_1} to y_{c_2} are exchanged.
- position exchange: genes from the parts y_0 to y_{c_1} and y_{c_2} to y_{q-1} are randomly selected to exchange their positions.

- **Order crossover (OX)**

The first stage is the same as in PMX, but, for the OX crossover strategy is:

- the parents crossing: the part y_{c_1} to y_{c_2} are exchanged.
- sliding motion: for the parent A and B, randomly selected places are set to *holes*, then sliding motion started from one of the crossover points is used to fill the holes as in figure 3.7.

- **Rotated Crossover (RX)**

For the RX proposed in this work, the parent crossing points could be one or more crossing points. The rotation operator we proposed is a closed loop shifting for genes in the chromosome for one or more positions. For example, with two-points crossover, parent A is rotated one-gene to the left while parent B is rotated one-gene to the right as in figure 3.8.

Applying a suitable mutation could enhance the children to update the population. One-point, two-points, randomized and deterministic mutation methods are proposed to be concatenated with simple, RX, PMX and OX crossover.

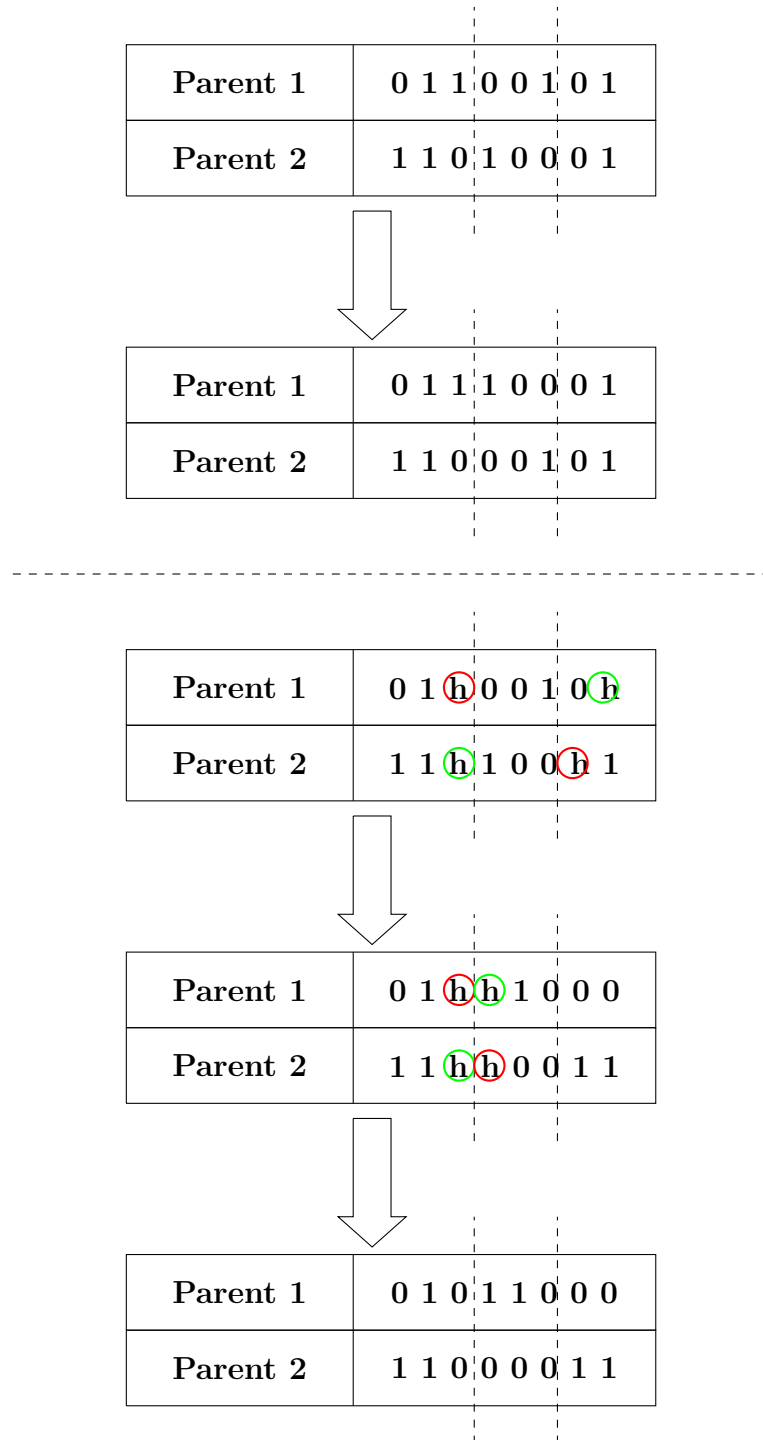


FIGURE 3.7: The order crossover

3.3.3.2 Mutation operators

To enhance the children, one or more genes could be randomly selected. If the new value of the selected gene is greater than the current value, the sum of all will be greater and so better solution. We have investigated four mutation operators

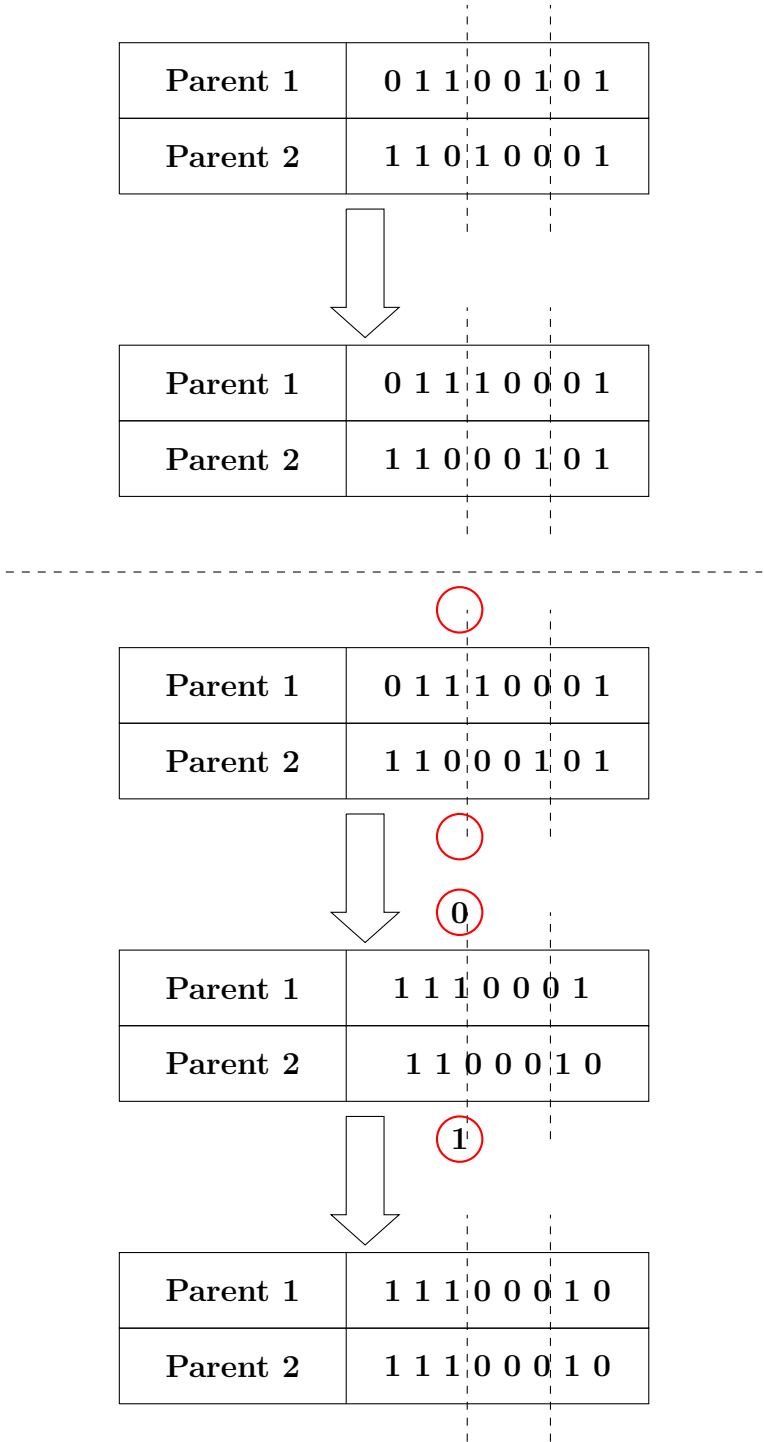


FIGURE 3.8: Rotated Crossover

and increasing strategies: one-point, two-points, deterministic and randomized. By merging each two of this four, we obtain the following mutations:

1. One-point deterministic
Randomly select one gene for mutation. The new value of the selected gene

is equal to the current value plus a constant value.

2. One-point randomized

Randomly select one gene for mutation. The new value of the selected gene is equal to the current value plus a random value.

3. Two-points deterministic

Randomly select two genes for mutation. The new value of the selected gene is equal to the current value plus a constant value.

4. Two-points randomized

Randomly select two genes for mutation. The new value of the selected gene is equal to the current value plus a random value.

Example 5: the GA illustration

Consider example 1 with $S = \{s_1, s_2, s_3\}$, $T = \{t_1, t_2, t_3, t_4\}$, $T(s_1) = \{t_1, t_2, t_4\}$, $T(s_2) = \{t_2, t_3, t_4\}$ and $T(s_3) = \{t_1, t_3, t_4\}$.

Before starting, the initial energy E_i and the energy cost of this period $E_i(k)$ for each sensor s_i are assumed to be known in addition to the monitoring season k .

Step 1: Sort the data to get all possible covers with the minimum number of sensors: $C_1 = \{s_1, s_3\}$, $C_2 = \{s_1, s_2\}$ and $C_3 = \{s_2, s_3\}$, considering that each of C_1 , C_2 and C_3 must cover all the targets in T .

Step 2: Assume that the period is k and the energy cost is $E_i(k)$, and let the initial energy be $10E_i(k)$. Therefore, sensor s_i and so cover c_l can be scheduled for 10 periods max as illustrated in equation 3.14. The number y_l of monitoring seasons could be scheduled for a cover $C_l \in C$ is bounded as $0 \leq y_l \leq E_i/E_i(k)$ which $0 \leq y_l \leq 10$ for $l = 1, 2, \text{ and } 3$ and the GA is implemented as follows:

1. The total available energy $30E_i(k)$ divided by the minimum period energy cost $2E_i(k)$ equal to 15 periods as an optimal possible solution.

2. Initialization

Randomly generate an initial population with a limited size of candidate schedule with covers' number of periods between 1 and 10 as in table II.

TABLE 3.1: Initial population

Schedules	y_1	y_2	y_3	Sum
1	2	3	4	9
2	4	7	5	16
3	6	4	8	18
4	4	5	7	16
5	5	4	3	12

3. Fitness and evaluation

When the energy consumption constraint is applied on the initial population as a fitness procedure, some of them can meet this constraint. Only schedule 1 and 5 met the constraint as in figure 3.9.a. The objective function is used to evaluate and select two parents. The candidate with the greatest sum (“max”) among those which satisfy the constraint are selected.

4. Crossover

Consider the two selected parents, then, crossover using the suitable number of points (single or multiple), i.e. generate a random number to identify the crossover point, then, crossover the chromosomes. Considering schedule 1 and 5 as parents, then run suitable single-point crossover as in figure 3.9.b for example.

5. Mutation

Again, generate a random number to identify the mutation point(s), then realize the mutation by increasing in a deterministic or random way the selected gene(s) to produce new enhanced children as in figure 3.9.c.

6. Children’s evaluation and population update

Regarding the energy consumption in equation 3.9, the fitness and evaluation are applied to the new enhanced children to update the parents if new enhanced children are better.

Consider the energy consumption constraint, repeat the crossover, mutation and Children’s evaluation in step 4, 5 and 6 until the sum of genes of a chromosome is equal to the found upper bound or the limit number of generations is reached.

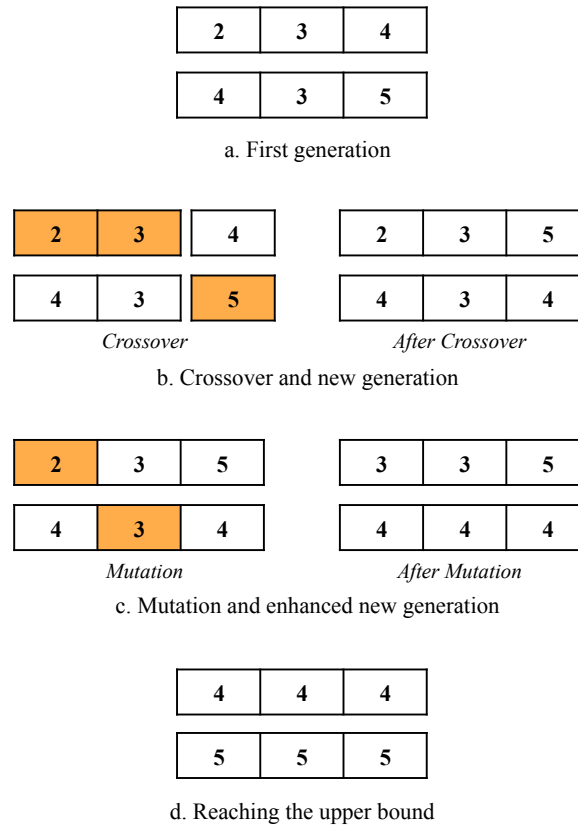


FIGURE 3.9: Genetic algorithm main stages.

Then, stop generating new chromosomes. Regarding the second chromosome in figure 3.9.d, the sum of genes is 15, which is equal to the upper bound. Therefore, no new generations of chromosomes are required.

3.3.3.3 GA configurations

Regarding the four crossover operators and the four mutation operators, the GA-based algorithm could be built using every crossover from the four, with every mutation operator from the four. Therefore, we have 16 different permutations and GA-based algorithms configurations, (see figure 3.10).

The Genetic Algorithm Configurations

Data:

Number of sensors :

Number of covers :

Number of generations :

Population size :

IR file name :

Initial Energy :

Energy Consumption Rates:

Crossovers

☒ One Point Crossover

☐ Two Points Crossover

☐ Partially Matched Crossover (PMX)

☐ Order Crossover (OX)

☐ Rotated Crossover (RX)

Mutation

☒ One Points Deterministic

☐ One Point Random

☐ Two Points Deterministic

☐ Two Point Random

FIGURE 3.10: GA Configurations

3.4 An existing DSC based method

Let us recall that the authors, in [83], proposed a GA-based method, called scattering, to be used for DSC maximization (GA-DSC), where the chromosome representation, selection, crossover, mutation, and operator were applied on the offspring.

The sensors were randomly distributed into groups as candidate covers, as in figure 3.11 .

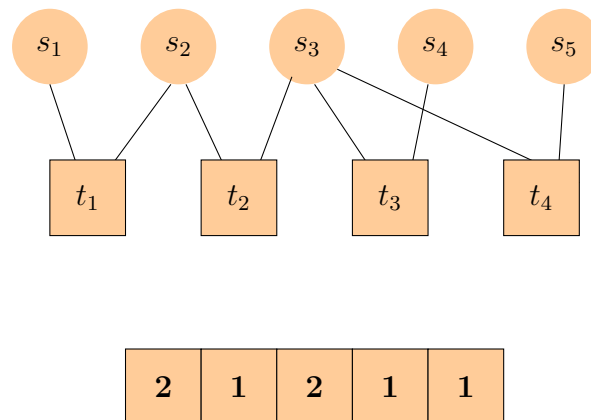


FIGURE 3.11: DSC encoding in GA.

1. Initially, each sensor randomly joins a group among prescribed groups. Then a group forms a cover if it can cover all targets. They considered that there is a limited number of DSC or upper bound “ ub ” related to the target t_j with the minimum $S(t_j)$. It is impossible to find more than ub disjoint covers if a target is only covered by $card(S(t_j)) = ub$ sensors as a minimum. Thus, the number of the prescribed groups is used as an upper bound ub of the number of covers.
2. The fitness of a chromosome is defined as the number of disjoint covers that can be found by the grouping combination represented by the chromosome.
3. Uniform crossover exchanges each gene of the two parents and mutation changes a randomly selected gene to a random value from 0 to ub to the new generation.

repeat the crossover, mutation and Children’s evaluation in step 2 and 3 until the number of DSC of a chromosome be equal to the found upper bound or the limit number of generations be reached.

3.5 Conclusion

This chapter provides an integrated vision for the proposed methods for WSNs lifetime optimization. It describes a method that contributes to solving the problem of WSNs lifetime optimization based on NDSCs. Also, it describes and uses an existing method from the literature to evaluate the proposed method. This problem is solved through two phases: by seeking the number of NDSCs and determining the optimal scheduling and utilization of the NDSCs that maximize the WSNs lifetime. We developed a binary representation based method and GA based method for the first phase. Then, we worked out the mathematical model for the WSNs lifetime scheduling and optimization for the second phase using an ILP model that could also be used to find the exact solution to this problem. Also, we developed a GA-based scheduling method to search the optimal solution

in reasonable execution time compared to the execution time required for the exact method. For a set S of sensors used to monitor a set T of targets, the first phase generates a set q of NDSCs. Then, the second phase has allowed to finding the optimal number of monitoring season for each cover y_l for all $l \in \{1, 2, \dots, q\}$, that maximize the WSNs lifetime. The proposed method aims to find the exact and optimal solution, using the integer linear programming mathematical model developed for this problem. For the near-optimal solution, we mentioned that we had developed a GA-based method with different crossover and mutation operators. The SX, OX, PMX, and RX are suggested to be implemented with the one-point, two-points, statistic and randomize mutation operators.

In the next chapter, we will describe the utilization of the proposed method to search for the solution of the WSNs maximum lifetime based on NDSCs. We will present the results of the experimental investigations obtained using these methods, as well as the evaluation of DSC-based method from the literature.

Chapter 4

Evaluation of the Methods through Numerical Simulation

4.1 Overview

This chapter describes the simulation environment used for all methods developed for WSNs lifetime optimization, comprising the ILP-based exact methods and the GA based heuristics. We describe briefly the experimental results for each method, with different instances. Then, the results obtained by the GA-based heuristics are compared to the exact method results obtained by the ILP on the same instances. Additional evaluation and comparison to GA based method for WSNs lifetime optimization represented into DSCs maximization (GA-DSC), as existing method from the literature, is briefly described. The previous chapter has shown that the problem of WSNs lifetime optimization is investigated using the NDSCs-based approach. The problem is split into two sub-problems: finding the optimal number of NDSCs and finding the optimal number of monitoring seasons for each cover to be implemented, to optimize the WSNs lifetime considering the available initial energy. As explained, the number of NDSCs aimed to be found using a binary representation method and GA based method. Then, the WSNs lifetime optimization problem has been designed to be solved using the ILP and

the GA-based heuristic. All these methods are programmed and applied on different instances so as to evaluate digital experimental results and the methods effectiveness. Then, the experiment results from numerical simulations are used for investigating the performances of the algorithms and quality of solutions, considering the instances and the capability of these algorithms to obtain the optimal solutions. This section describes the environment and the raw data used for all algorithms implementation regarding the machine and the programming languages. All methods were implemented on HP Elite Book 8770w work-station with Intel CORETMi7 2.70GHz processor, 16GB RAM, and Microsoft windows7 -64bit operating system. The proposed methods have been programmed using C language environment. For the programs execution, the primary scheduling data (the number of sensors, the number of covers, the set cover matrix V , the initial energy E_i and the energy consumption $E_i(k)$ for a given period k) are required. Also, the linear optimization algorithm was implemented in Eclipse development environment and using the Cplex linear optimization libraries and the Java Development Kit (JDK).

Before implementing the algorithms, it is necessary to explain that the input data including the sensors deployment and the targeted positions are randomly generated. Regarding a sets of sensors composed of $\{5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500\}$, there are (x_i, y_i) of each sensor $s_i \in S$ position and the coverage range d_i of these sensors. The random data for sets of sensors are saved in files.

4.2 The binary representation method for finding the NDSCs

Initially, the sensor network is deployed randomly to monitor targets as follows:

1. *Sensors deployment and targets positioning*

In this stage, each sensor from S and target from T are located in a two-dimensional space by assignment of a random value to the pair elements

(x, y) in $10 \times 10l.unit$ for example.

2. Creation of the individual cover relations matrix Δ

Based on the sensors and targets locations and the coverage range of each sensor $s_i \in S$, the matrix Δ is constructed according to targets j from T in the coverage area of sensor s_i from S .

$$\Delta = (\delta_{ij}) = \begin{pmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \delta_{m1} & \delta_{m2} & \dots & \delta_{mn} \end{pmatrix}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if sensor } i \text{ can cover target } j \\ 0 & \text{otherwise} \end{cases}$$

3. Finding all possible covers and construct the matrix V

To find all possible sensors distribution into covers, we consider all the possibilities of each sensor i in S to be included in the $2^m - 1$ possible covers, as in W matrix.

$$W = (w_{ij}) = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_{(2^m-1)1} & v_{(2^m-1)2} & \dots & v_{(2^m-1)m} \end{pmatrix}$$

Then, search for each of these candidate possible covers in W matrix that can monitor all the targets in T to be selected into V .

$$V = (v_{li}) = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_{q1} & v_{q2} & \dots & v_{qm} \end{pmatrix}$$

$$v_{li} = \begin{cases} 1 & \text{if sensor } i \text{ is included in cover } l \\ 0 & \text{otherwise} \end{cases}$$

q : the number of possible covers

4.3 The GA based method for finding the NDSCs

The algorithms were coded using the C programming language for experimental study. The programs were implemented on a randomly deployed set of sensors ($S = \{10, 20, 30, \dots, 500\}$), and used to monitor a set T of targets. We investigated different numbers of generations $ng \in \{1, 2, \dots, 30\}$ with different population sizes $ps = \{100, 200, 300, 400, 500, 600\}$. The table 4.1 presents the number of NDSCs found using different numbers of sensors used to monitor a set T of targets $n = 5$. The numbers ng of generations is equal to 20, and the population size is equal to 500.

TABLE 4.1: The relation between the population size and number of NDSCs

Number of Sensors	10	20	30	40	50
Number of NDSCs	561	1603	2059	2239	2979

For the study of the effect of the sizes of the populations, figure 4.1 shows different

numbers of generations with a constant number of targets and sensors equal to 5 and 10 with population size equal to 200 and 500.

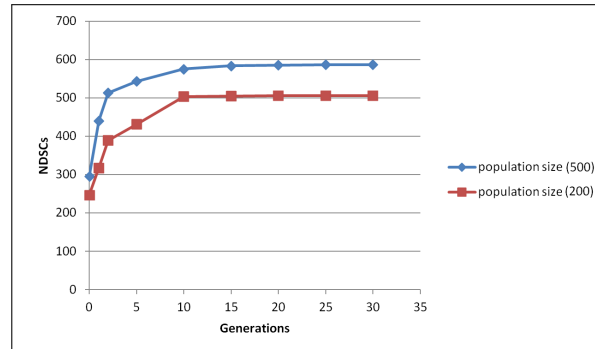


FIGURE 4.1: The number of cover for small and greater population size

The experimental results brought out the significance of the population size effect on the results obtained with the same number of generations. The Table 4.2 highlights this effect using number of generation equal to 20.

TABLE 4.2: The relation between the population size and number of NDSCs (ng=20)

Population Size	100	200	300	400	500	600
Number of NDSCs	374	506	591	647	687	716

Considering the extensibility, it is now possible to find the NDSCs for the larger number of sensors. The Table 4.3 shows the number of covers for 50 to 500 sensors, the number of generations and population size are equal to 20 and 500 respectively.

TABLE 4.3: The WSN expandability

Number of Sensors	50	100	200	300	400	500
Number of NDSCs	2979	3924	5443	6650	7781	9013

Regarding the execution times, both, the number of sensors and the number of generations should be considered. The Table 4.4 shows the execution time in milliseconds for the number of sensors respectively equal to 100, 200, 300, 400 and 500 with a number of generations equal to 500, 1000, 1500 and 2000.

TABLE 4.4: GAMSC and GANDSC in ms

Num. of generations	Num. of Sensors				
	100	200	300	400	500
500	3830	15091	35274	68535	111176
1000	7568	33064	66793	136534	233428
1500	11789	44259	96843	226906	351211
2000	15883	58241	150573	301891	470178

The mathematical model of this scheduling problem of the covers is formulated through integer linear programming to find out an exact solution. Combining the exact method and the genetic algorithm heuristic allowed us to reach the optimal lifetime and to evaluate the effect of NDSC based method on the WSN lifetime. Then, we compared the results to DSC-based GA from the literature for evaluation. Using the NDSC based GA, we obtain the number q of covers that must be scheduled to maximize the WSNs lifetime.

4.4 ILP based method simulation and results

Considering the number q of NDSCs obtained using either the binary representation method or the GA based method, the optimal lifetime of the WSNs is achieved by finding the optimal utilization of the NDSCs. The optimal scheduling aims at finding the optimal number of monitoring seasons y_l for each cover $C_l \in C$. As mentioned in the previous chapter, the solutions of both the exact algorithms and the heuristics are targeted. This subsection explains how to obtain the exact solution for this problem, through the ILP model and using Cplex. Numerical experiments have been carried out on random data based networks with $card(S) \in \{5, 10, 20\}$ and $card(T) = 5$. Considering the initial energy $E_i = 10, 20, \dots, 60$ and energy consumption on all sensor for a period k $E_i(k) = 2, 4, 8, 16$, fourteen instances have been investigated. Eclipse supported

software program with the linear optimization libraries of Cplex and Java Development Kit (JDK) were used to design the linear optimization algorithm, using the computation resources and the environment described before. Experimental results are obtained for WSNs with different numbers of sensors and different energy situations. For $S = \{s_1, s_2, s_3, s_4, s_5\}$ and $T = \{t_1, t_2, t_3, t_4, t_5\}$ randomly deployed in 10×10 area, that is $m = 5$ and $n = 5$, the coverage relations matrix is

$$\Delta = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

There are $2^5 - 1 = 31$ possible arrangements of the sensors into covers distribution in the $tembV$ matrix. Only 15 of them are covers as in the cover relations matrix as:

$$V^T = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Therefore, the objective function from equation 4 becomes:

$$L = k \times \sum_{l=1}^{15} y_l \quad (4.1)$$

Subject to :

$$\sum_{l=1}^{15} E_i(y_l)v_{il} \leq E_i \quad i = 1, 2, \dots, 5 \quad (4.2)$$

One seeks to maximize an objective function subjected to 5 inequality constraint equations involving 15 unknowns. Therefore, this problem can be formulated linearly as follows:

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} \quad (4.3)$$

Subject to :

$$y_1 + y_2 + y_4 + y_6 + y_7 + y_8 + y_9 + y_{11} + y_{13} + y_{15} \leq E_1/E_1(k)$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_9 + y_{10} + y_{11} + y_{14} + y_{15} \leq E_2/E_2(k)$$

$$y_2 + y_5 + y_6 + y_8 + y_9 + y_{12} + y_{13} + y_{14} + y_{15} \leq E_3/E_3(k)$$

$$y_3 + y_4 + y_5 + y_6 + y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} \leq E_4/E_4(k)$$

$$y_7 + y_8 + y_9 + y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} \leq E_5/E_5(k)$$

For all cover $l \in \{1, 2, \dots, 15\}$, y_l are bounded as:

$$0 \leq y_l \leq E_i/E_i(k) \quad l = 1, 2, \dots, 15 \quad (4.4)$$

Applying this enhancement and simplification method on the matrix V , each row covers completely the targets. Therefore, one can find the covers with the minimum number of sensors. The sum of each row in V gives the number of sensors in this cover. So, finding the cover with minimum sensors amounts to finding the row with minimum sum. When a cover is selected, this cover can't be included in next covers, which discards covers that include small size covers as the previous lemma 2.

1) For $S = \{s_1, s_2, s_3, s_4, s_5\}$ and $T = \{t_1, t_2, t_3, t_4, t_5\}$ randomly deployed in 10×10 area, that is $m = 5$ and $n = 5$, as in example 2, the cover relations matrix V becomes

$$V = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The objective function from equation 4.1 and the inequality constraint in equation 4.2 become:

$$L = k * \sum_{l=1}^4 y_l \quad (4.5)$$

Subject to :

$$\sum_{l=1}^4 E_i(y_l) V_{il} \leq E_i \text{ for all } i = 1, 2, \dots, 5 \quad (4.6)$$

Here, one seeks to maximize an objective function subjected to 5 inequality constraint equations with 4 unknowns instead of 15. Therefore, this problem can be formulated linearly as follows:

$$y_1 + y_2 + y_3 + y_4 \quad (4.7)$$

Subject to :

$$E_1(y_1) + E_1(y_3) \leq E_1$$

$$E_2(y_1) + E_2(y_2) \leq E_2$$

$$E_3(y_3) + E_3(y_4) \leq E_3$$

$$E_4(y_2) + E_3(y_4) \leq E_4$$

$$E_5(y_3) + E_5(y_4) \leq E_5$$

For all cover $l \in \{1, 2, 3, 4\}$, y_l are bounded as

$$0 \leq y_l \leq E_i / E_i(k) \quad l = 1, 2, 3 \text{ and } 4$$

Then, the simplified and integer linear scheduling and optimization strategy (SILSOM) applied and compared to integer linear scheduling and optimization strategy (ILSOM) for the same instances. The Table 4.5 shows the results provided by both integer linear scheduling and optimization method ILSOM and the simplified integer linear scheduling and optimization method SILSOM. It is clear that the enhancement has simplified the problem without affecting the optimal solutions.

2) For $S = \{s_1, s_2, \dots, s_{10}\}$ and $T = \{t_1, t_2, t_3, t_4, t_5\}$ randomly deployed in 1×10

TABLE 4.5: SILSOM and ILSOM

Sensors Network			SILSOM						ILSOM	
Sensors	E_i	$E_i(k)$	Covers	C_1	C_2	C_3	C_4	Opt. ($*k$)	Covers	Opt. ($*k$)
5	160	2	4	0	80	80	0	160	15	160
5	160	4	4	0	40	40	0	80	15	80
5	160	8	4	0	20	20	0	40	15	40
5	160	16	4	0	10	10	0	20	15	20

area, that is $m = 10$ and $n = 5$ as in example 2, the cover relations matrix is

$$CR = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

There is $2^{10} - 1 = 1023$ possibilities of sensors sets distribution in the W matrix, but only 784 are covers according to the binary representation method. Therefore, this problem is formulated linearly with an objective function, 10 inequality constraint equations, and 784 unknowns. This problem is solved and the optimal lifetime for different energy situations are shown in Table 4.6 according to the ILSOM method.

TABLE 4.6: ILSOM for 10 sensors

Sensors Network			ILSOM	
Sensors	E_i	$E_i(k)$	Covers	Optimal ($*k$)
10	10	2	784	15
10	20	2	784	30
10	30	2	784	45
10	40	2	784	60
10	50	2	784	75

Using the SILOM for the same 10 sensors, the objective function can be represented as the sum of 26 unknowns

$$\sum_{l=1}^{26} y_l \quad (4.8)$$

In this case, the constraints reduce to 10 inequalities with 26 unknowns such as:

$$y_1 + y_3 + y_7 + y_{11} + y_{14} \leq E_i/E_i(k)$$

$$y_1 + y_2 + y_5 + y_{10} + y_{16} + y_{18} + y_{22} \leq E_i/E_i(k)$$

$$y_3 + y_4 + y_6 + y_{23} \leq E_i/E_i(k)$$

$$y_2 + y_4 + y_8 + y_{10} + y_{12} + y_{15} + y_{24} \leq E_i/E_i(k)$$

$$y_3 + y_4 + y_{11} + y_{12} + y_{19} + y_{24} \leq E_i/E_i(k)$$

$$y_5 + y_6 + y_9 + y_{13} + y_{20} \leq E_i/E_i(k)$$

$$y_7 + y_8 + y_9 + y_{14} + y_{15} + y_{17} + y_{21} + y_{25} \leq E_i/E_i(k)$$

$$y_{10} + y_{11} + y_{12} + y_{13} + y_{14} + y_{15} + y_{18} + y_{19} + y_{20} + y_{26} \leq E_i/E_i(k)$$

$$y_{16} + y_{17} + y_{18} + y_{19} + y_{20} + y_{21} \leq E_i/E_i(k)$$

$$y_{22} + y_{23} + y_{24} + y_{25} + y_{26} \leq E_i/E_i(k)$$

Considering the initial energy E_i and the energy consumption $E_i(k)$, y_l cannot be more than $E_i/E_i(k)$. This can be bounded as

$$0 \leq y_l \leq E_i/E_i(k)$$

This model is very simple, compared with a model composed of an objective function and 10 inequality constraints with 784 unknowns.

3) For $S = \{s_1, s_2, \dots, s_{20}\}$ and $T = \{t_1, t_2, t_3, t_4, t_5\}$ randomly deployed in $10x \times 10$ area, that is $m = 20$ and $n = 5$ as in example 2, there are $2^{20} - 1 = 1048576$ possibilities of sensors sets arrangements, and 910336 of them are covers according to definition 1. This problem is investigated and the optimal lifetime for different energy situations are shown in table 4.7 according to ILSOM method.

TABLE 4.7: ILSOM for 20 sensors

Sensors Network			ILSOM	
Sensors	E_i	$E_i(k)$	Covers	Optimal ($*k$)
20	10	2	910336	15
20	20	2	910336	30
20	30	2	910336	45
20	40	2	910336	60
20	50	2	910336	75

The exponential growth of the ILP model and the complexity of the problem when the number of sensors increases in table 4.8 makes the WSNs expansion unreachable. The computational efforts considering the resources and the execution time are limited to find the exact solution for a greater number of sensors.

TABLE 4.8: ILP model computation time samples

Number of sensors	number of covers	Execution time
5	15	0.02
20	910336	6.02

The execution time increased from 0.02 to 6.04 seconds (302 times) when the number of sensors increased from 5 to 20.

4.5 GA based method simulation and results

The GA-based method is used for its heuristics capability to yield an acceptable solution in a reasonable time in addition to its efficiency. The GA is used to maximize the WSNs lifetime, starting from the deployment data and targets positions and by searching the NDSCs. The C language program runs the GA for a given number of generations to determine the maximum number of monitoring seasons for each NDSC, corresponding to the lifetime. This subsection shows the results obtained using the GA with different crossover and mutation operators on various instances.

As in the Table 4.9, a WSNs with 10 sensors monitoring 5 targets are deployed randomly in a 10×10 area. The initial energy of each sensor is E_i unit and the energy consumption for a specified period k unit of time is $E_i(k)$. The possible number of covers found is 644. Each cover C_l can be scheduled for *zero* or more monitoring seasons y_l and the sum of scheduled monitoring seasons for all covers are laid out in the column “lifetime”, in the table. The lifetime can be calculated by multiplying the value in this column by k as mentioned earlier. The Table 4.9 displays the maximum lifetime for 160 units initial energy with different rates of energy consumption. We tested the GAs based on the following strategies: the GA11 (single point crossover, and single point mutation), the GA12 (single point crossover, and dual point mutation), the GA12 (dual point crossover, and single point mutation), the GA21 (dual point crossover, and dual point mutation), and the GA22 (deterministic and randomized mutation). While the population size can affect the efficiency and performance of GA [72], small size population of 50 to 1200 generations and larger size population of 100 to 1200 generations were investigated.

TABLE 4.9: Different GA mutation strategies

Network			Deterministic Mutation										Random Mutation					
			Population Size=50					Population Size=100					Population Size=100					
			Lifetime = (* k)					Lifetime = (* k)					Lifetime = (* k)					
m	E_i	$E_i(k)$	GA11	GA12	GA21	GA22	Av	GA11	GA12	GA21	GA22	Av	GA11	GA12	GA21	GA22	Av	
10	160	2	310	279	308	296	298	307	284	296	270	289	302	305	287	283	294	
10	160	4	153	150	148	135	147	153	154	157	147	153	154	157	150	155	154	
10	160	6	99	94	103	95	98	101	92	101	92	96.6	96	103	103	100	101	
10	160	8	74	76	77	73	75	77	77	75	71	75	75	80	70	76	75	
10	160	10	59	64	57	64	61	62	58	62	59	60	60	58	62	63	61	

A simple comparison is made between the mutation strategies regarding the number of generations and the closeness of the solution to the optimal. Considering the one point mutation, GA11, and two points mutation, GA12, the investigation have shown that the GA12 gives better results than GA11, for the same number of generations, as depicted in figure 4.2. The GA12 takes 1250 generations to obtain the optimal solution while GA11 takes 2250 generations.

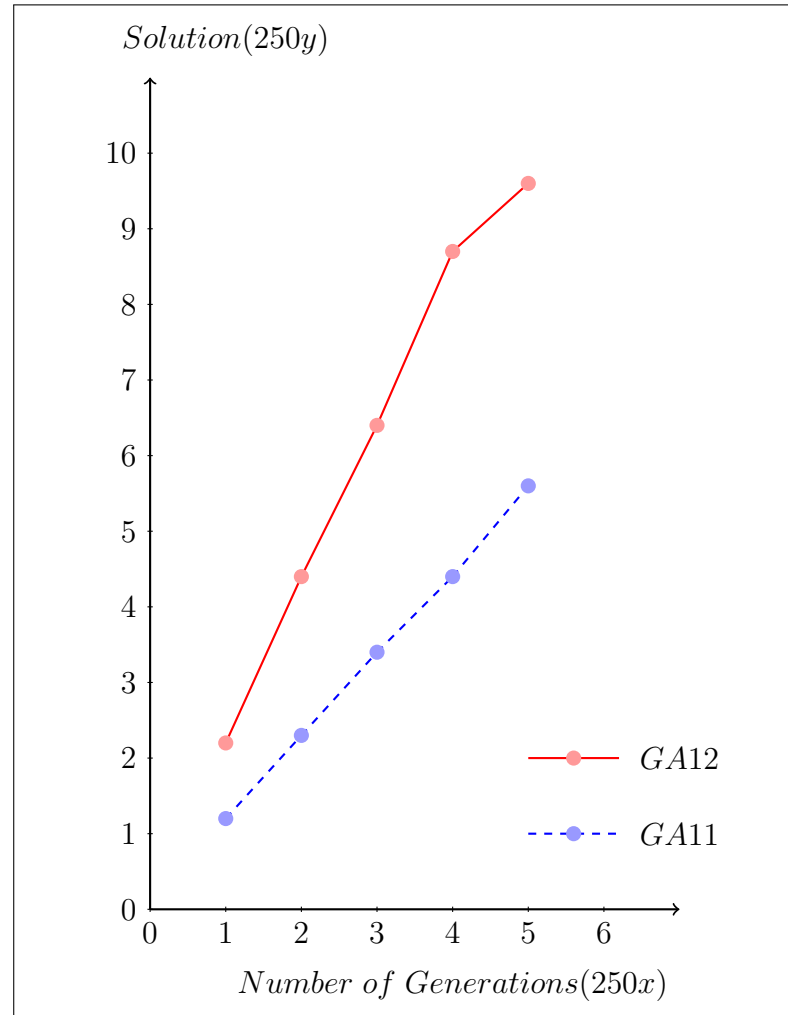


FIGURE 4.2: GA11 vs GA12.

Considering the deterministic and randomized mutation strategies, we have made a simple comparison between them depending on the number of generations and the closeness of the solution to the optimal. It is clear that the randomized GA11 gives better results than the deterministic GA11 for the same number of generations as

in figure 4.3. The deterministic GA11 takes 1250 generation to obtain the optimal while the randomized GA11 takes 2250.

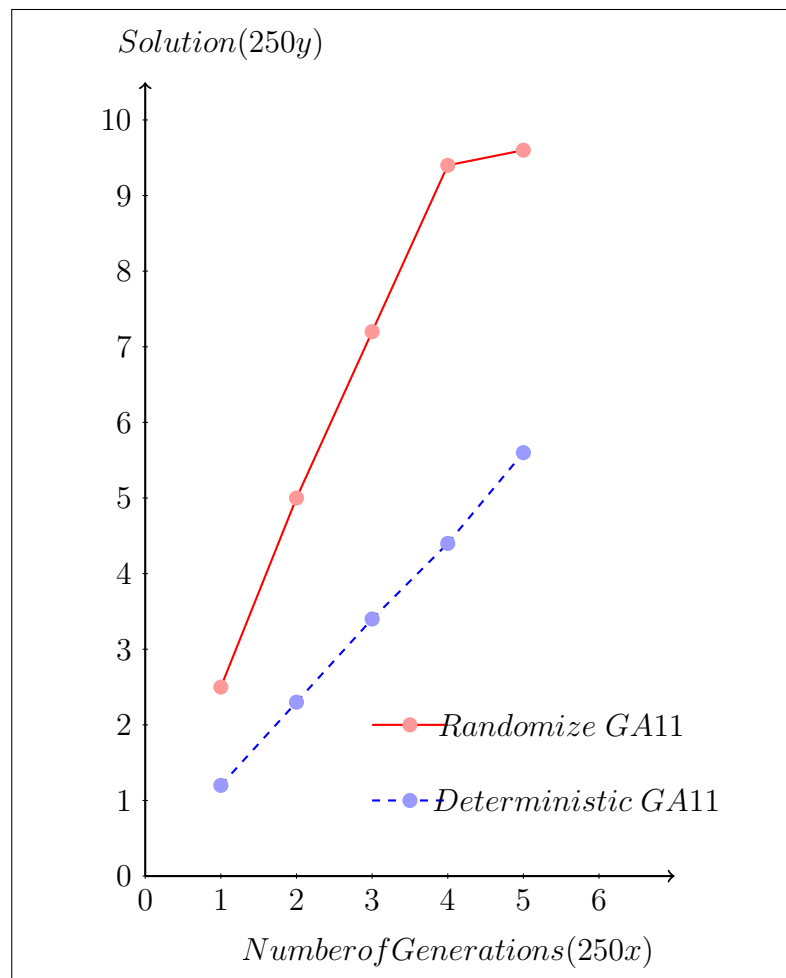


FIGURE 4.3: Deterministic vs Randomize GA11.

It is clear from these results that the performance of the GA is not critically dependent on the strategy of crossover and mutation when a large number of generations used "more than 2500 for example". Indeed, one can observe that the single mutation provides the best results for a small size population, while the results by the randomized mutation give greater lifetime than the deterministic mutation, for a large size population as mentioned in table 4.9.

Considering the effect of the great number of generations on the execution time, the number of generations with a reasonable execution time that provides an acceptable solution can be found for all strategies. The number of generation required for GA11 to get the optimal is not less than 2250 and 1250 for GA12. In addition,

other ways of crossover and mutation can be investigated in addition to different ways of initializing and generating the chromosomes for more enhancements, in order to choose the best strategy for this problem.

With referring to the crossover operators: SX, OX, PMX and RX mentioned in chapter 3, an investigation is made to assess the performance of each operator.

Using the binary representation method, $q = 21$ covers were found from $m = 10$ sensors. The lifetime optimization is investigated using the SX, PMX, RX and OX crossovers, for small population size ($p = 10$). The initial energy is assumed to be 60 units, and the energy consumption rate is 2 units for the 10 sensors. Figure 4.4 points out the efficiency of the SX, PMX, RX and OX crossovers to optimize the lifetime with a number of generations equal to 50, 100, 150 and 200 respectively.

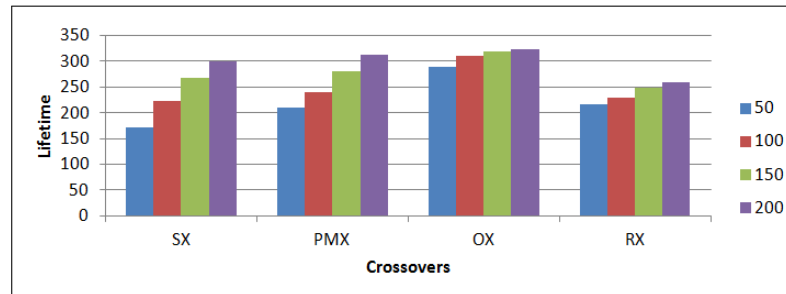


FIGURE 4.4: The WSNs lifetime using SX, PMX, RX and OX crossover with $p=10$.

To study the effect of the population size using the SX, PMX, RX and OX crossovers, we used a greater population size ($p = 100$), for the same instances used for figure 4.4, and the lifetime obtained is displayed in figure 4.5.

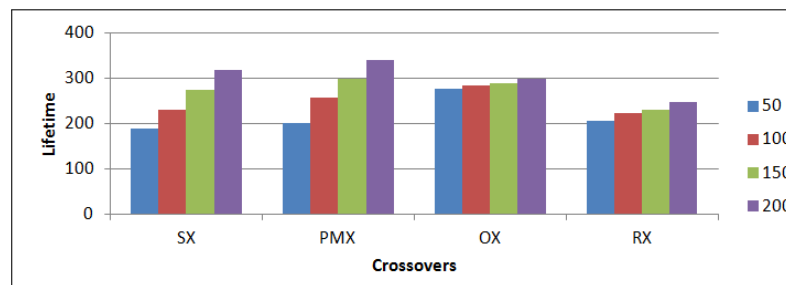


FIGURE 4.5: WSNs lifetime using SX, PMX, RX and OX crossover with $p=100$.

We used different mutation strategies jointly with the simple, PMX, RX and OX crossovers, to implement different GA algorithms. The lifetime obtained using the

SX, PMX, RX and OX crossovers, jointly with one point or two points mutation, for the same instances, is presented in figure 4.6.

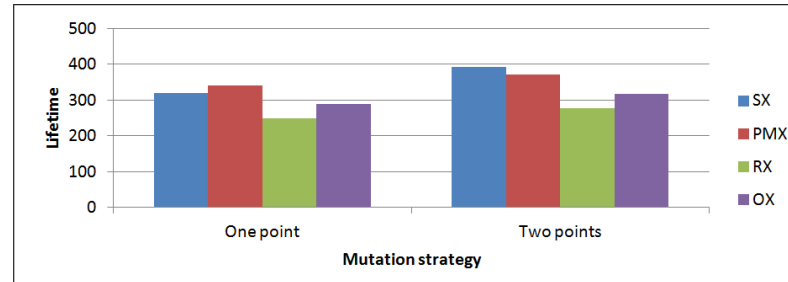


FIGURE 4.6: The lifetime using one and two points mutations.

The lifetime obtained using the SX, PMX, RX and OX crossovers, jointly with one point deterministic and randomized mutations for the same instances is presented in figure 4.7.

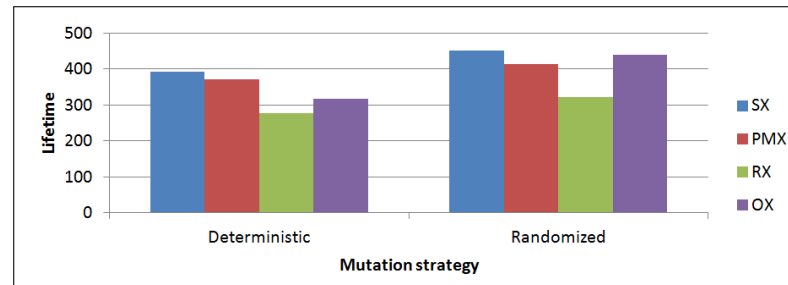


FIGURE 4.7: The lifetime using deterministic and randomized mutations.

Then, the energy consumption for all sensors used in the covers are investigated with all crossover and mutation operators for different instances. The energy consumed by solutions obtained using simple crossover for different numbers of generations $\{10, 100, 1000\}$ is presented in figure 4.8 considering 10 sensors with initial energy equal to 60 units.

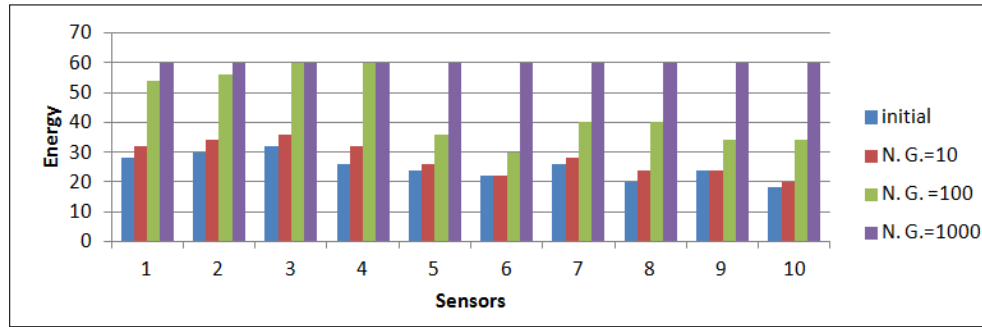


FIGURE 4.8: The energy consumption via all sensors.

The covers utilization by solutions using simple crossover for different number of generations is presented in figure 4.9.

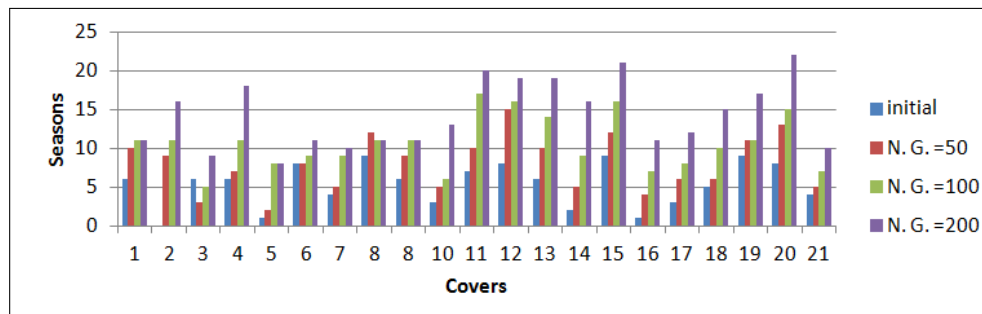


FIGURE 4.9: The 21 covers utilization via different number of generations.

The energy consumed by solutions obtained using the SX, PMX, RX and OX crossovers, for a number of generations equal to 200 is presented in figure 4.10 considering 10 sensors with initial energy equal to 60 units.

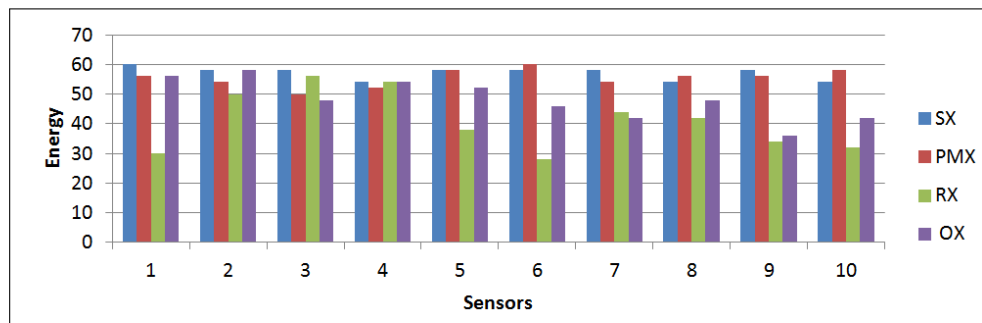


FIGURE 4.10: Energy consumption by all sensors using SX, PMX, RX and OX crossovers.

The covers utilization via solutions obtained using the SX, PMX, RX and OX crossovers, for a number of generations equal to 200 is presented in figure 4.11.

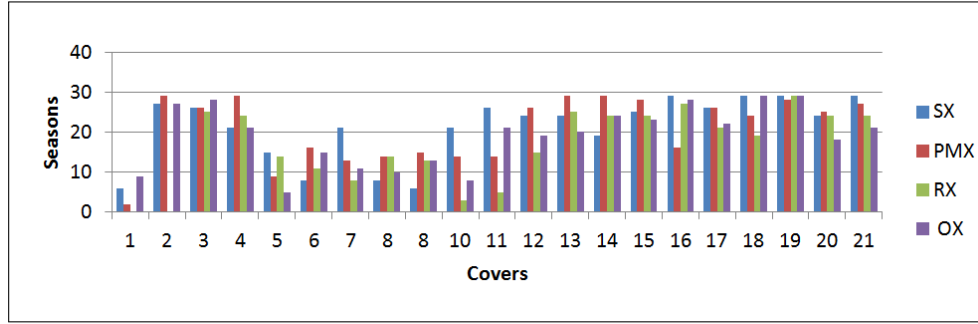


FIGURE 4.11: The 21 covers utilization via different number of generations using SX, PMX, RX and OX crossovers.

Then, we investigate the opportunity of implementing the algorithms for a WSN composed of sensors with different initial energy and different energy consumption rate. For a WSN with 10 sensors, with initial energy $E_i = \{60, 110, 150, 80, 72, 130, 50, 120, 90, 100\}$ and the energy consumption rate $E_i(k) = \{2, 5, 3, 4, 6, 5, 2, 4, 3, 5\}$, the optimal lifetime obtained by all the algorithms is presented in figure 4.12.

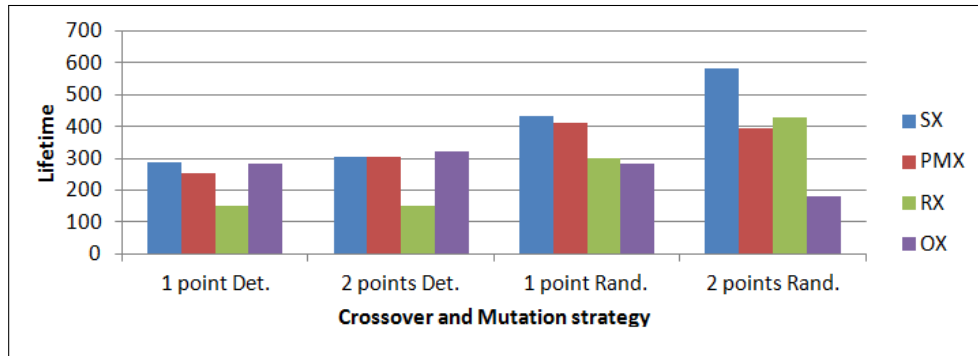


FIGURE 4.12: The lifetime for WSN with not identical sensors.

We have also investigated the performance of the algorithms we developed for WSNs lifetime optimization, based on NDSCs, considering both NDSCs search and scheduling. We obtained the optimal solution for this problem using the ILP mathematical model we developed by exploiting either the binary representation method or GA-based method for the NDSCs search. The next subsection explores the result obtained by the DSC-based GA from the literature to be compared with the results we achieved with our algorithms.

4.6 The existing method simulation and results

The GA based method used of WSNs lifetime optimization represented into DSCs maximization problem developed in [83] is programmed using C programming language. The results obtained using the method is compared against our results. For a number of sensors ($m = \{10, 50, 100, 200, 300, 400, 500\}$), the upper bound for the number of DSCs and the optimal solutions obtained are depicted in figure 4.13 with the number of iterations equal to 100.

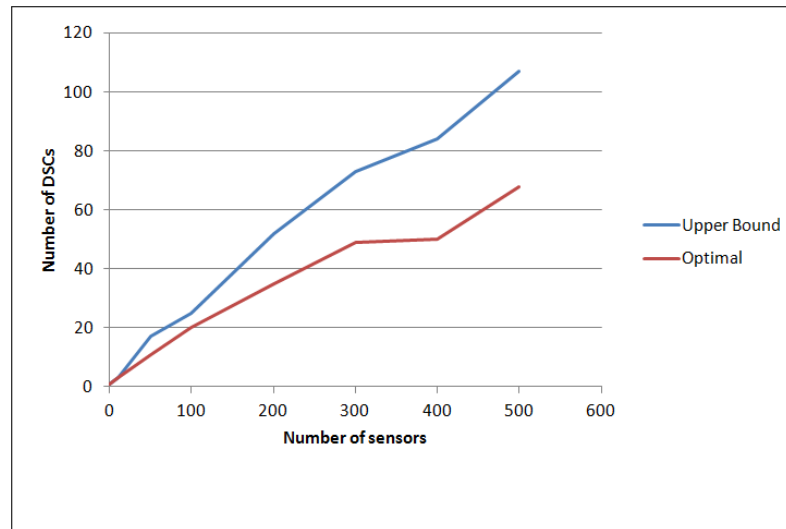


FIGURE 4.13: The DSCs for different numbers of sensors.

For a number of sensors equal to 500, the enhancement of the solution by increasing the number of iterations is depicted in figure 4.14. The upper bound for the number of DSCs is 107, and the near optimal obtained is 101 in 10000 iterations.

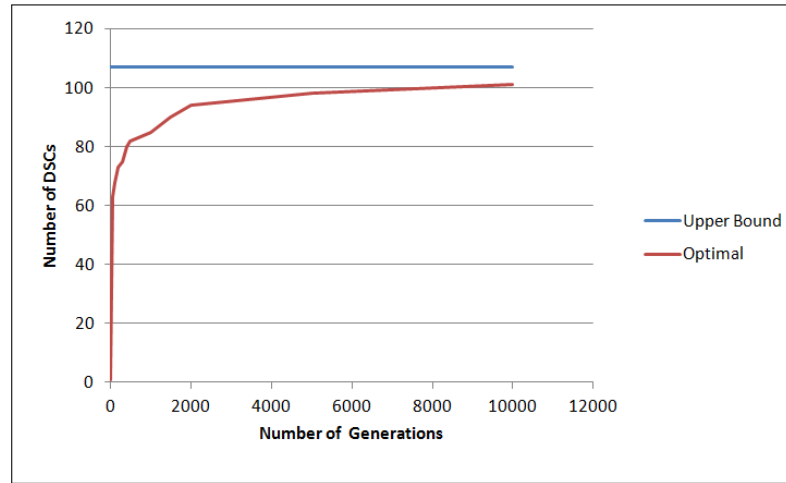


FIGURE 4.14: The DSCs for different numbers of iterations.

4.7 Results analysis and evaluation

Based on the randomly deployed sensors and targets, the binary representation method and the GA based method were used to generate the maximum possible number q of covers. Then, these covers were scheduled using the GA based heuristic methods with different strategies (NDSCGA) and the ILP method to find a near-optimal heuristic solution and optimal solution respectively for the WSNs lifetime optimization. Again, the GA-based method used in the literature to find the optimal solution for the WSNs lifetime formulated as DSC maximization problem. The analysis and evaluation of the related results could be presented and evaluated at three levels: 1) the different GA-based methods that we developed, for NDSC maximization (NDSCGA) and used for WSNs lifetime optimization, have been assessed and compared to each other, 2) the NDSCGA heuristic solutions have been evaluated and compared to the ILP based exact solutions and 3) the NDSCGA heuristics solutions have been assessed and compared to DSC based GA (DSCGA) from the literature. The following subsections analyze the results.

4.7.1 Evaluating the GA based strategies results

This subsection investigates the different crossover operators used in the different GA: SX, CX, OX and PMX implemented jointly with the mutation operators: one point, two points, deterministic and randomized. To investigate the enhancements of the crossover operators regarding the number of iterations, we used 10 non-identically loaded sensors with an initial energy vector ($E_i = \{60, 110, 150, 80, 72, 130, 50, 120, 90, 100\}$) and a energy consumption rate vector ($E_i(k) = \{2, 5, 3, 4, 6, 5, 2, 4, 3, 5\}$). The crossover operators are evaluated, jointly with the different mutation operators. The results obtained using all crossover operators with the one-point randomized mutation for a number of iterations up to 500 are shown in figure 4.15.

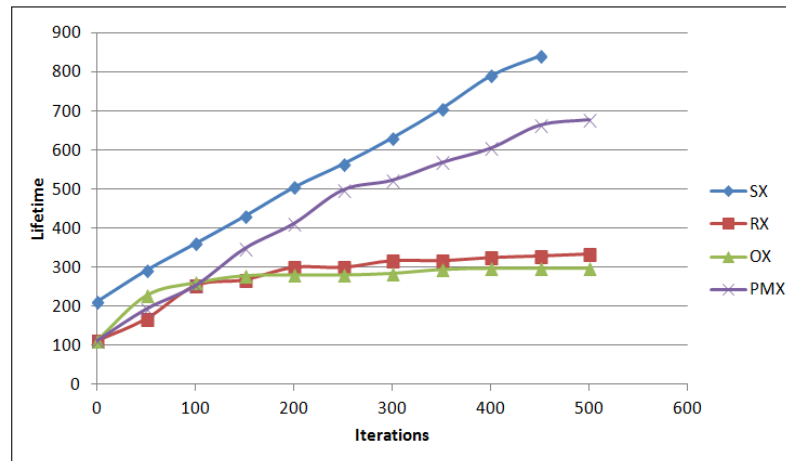


FIGURE 4.15: Crossover one-point randomized with different number of iterations.

The results obtained using all crossover operators, jointly with the two-point randomized mutation for a number of iterations up to 500 are shown in figure 4.16, for 10 non-identical sensors.

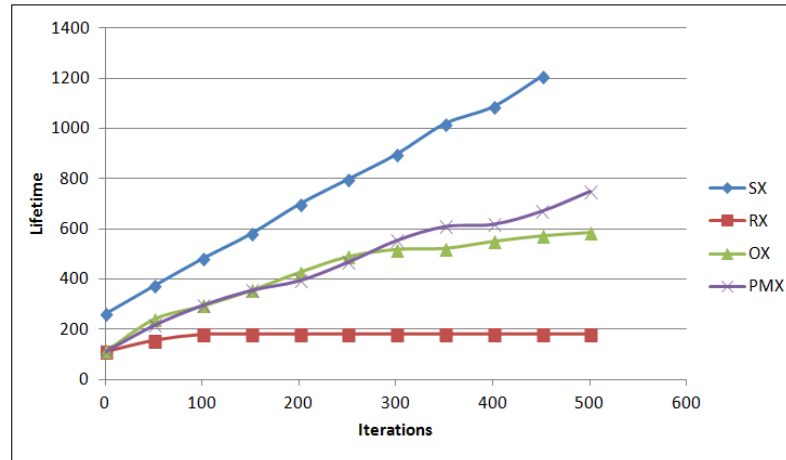


FIGURE 4.16: Crossover two-point randomized with different number of iterations..

The results obtained (with 10 non-identical sensors are used) using all crossover operators with one point deterministic mutation for a number of iterations up to 500 are shown in figure 4.17.

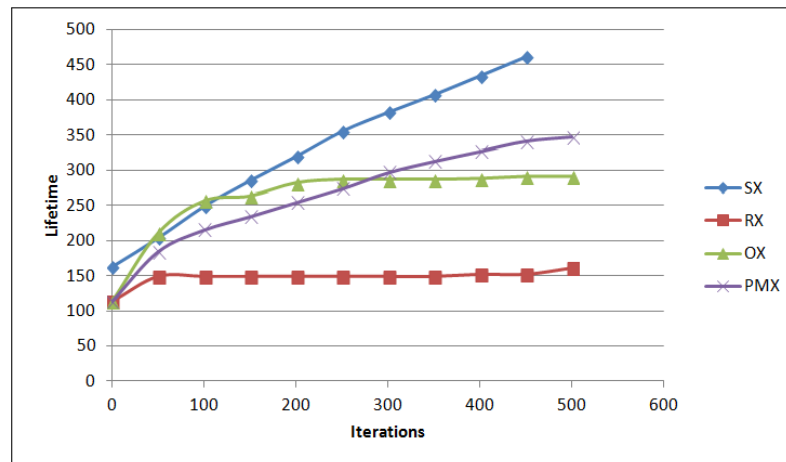


FIGURE 4.17: Crossover one-point deterministic with different number of iterations..

The results obtained (with 10 non-identical sensors are used) using all crossover operators with two-point deterministic mutation for a number of iterations up to 500 are shown s in figure 4.18.

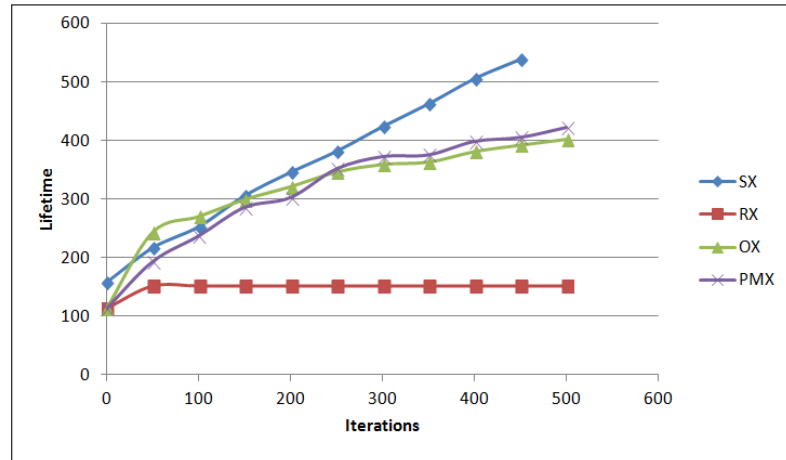


FIGURE 4.18: Crossover two-point deterministic with different number of iterations..

4.7.2 Evaluating the GA to the Optimal

In this subsection, the solution obtained using our GA-based method is compared to the exact solution obtained using the ILP considering the quality of the solutions and the execution times. The Table 4.10 highlights a simple comparison between the GA-based method and the ILP-based method using eight instances with $S \in \{5, 10\}$ and energy consumption cost $E_i(k) \in \{2, 4, 8, 16\}$, coverage range $r = 3$ and initially charged with 160 energy units used to monitor 5 targets. The execution time in seconds and the optimal lifetime as multiples of k are obtained using both methods for all instances.

TABLE 4.10: GA and ILP based mettods comparison

Instances	Lifetime = (optimal value * k)			
	ILP		GA	
$S, T, r, E_i, E_i(k)$	Ex. time (s)	Opt. value	Ex. time (s)	Opt. value
5, 5, 3, 160, 16	0.03	20	0.010	20
5, 5, 3, 160, 8	0.02	40	0.010	40
5, 5, 3, 160, 4	0.02	80	0.010	80
5, 5, 3, 160, 2	0.02	160	0.010	160
10, 5, 3, 160, 16	0.02	30	0.051	30
10, 5, 3, 160, 8	0.02	60	0.041	59
10, 5, 3, 160, 4	0.02	120	0.055	117
10, 5, 3, 160, 2	0.02	240	0.068	238

Consider the execution time for both methods and the closeness of the lifetime reached by the solution provided by the heuristics based on the GAs, the experimental results support the followings observations:

- It is possible to find the optimal solution using all suggested GA-based strategies, but with different numbers of generations and so different execution time.
- It is possible to extend the WSN size and lifetime by increasing the number of sensors, considering the increase of the execution time. The execution time rose from 0.02 to 6.04 seconds while the number of sensors has increased from 5 to 20.
- It is the same for the linear programming model execution time, which rose from 0.02 to 2.53 seconds when the number of sensors increased from 5 to 20. Therefore, the total execution time rose from 0.04 to 8.57 seconds when the number of sensors increased from 5 to 20.

- Additional time is required for the WSN random deployment and sorting to build the matrix V .
- Our new encoding method and mathematical model can be used as a basis for more advanced investigations to enhance the performance of the GA-based method.
- Regarding the optimal solution obtained using GA-based heuristic and the exact solution values, the GA-based heuristic has found the optimal solution in 5 out of 8 instances, which is greater than 50% ". The worst case is 117 of 120 which is 97.5%.
- Our linear programming based method can be used as an exact method to evaluate new heuristics for the randomly deployed WSNs lifetime maximization problem.

4.7.3 Evaluating the NDSCGA to the DSCGA

In the GA-based method proposed for WSNs lifetime optimization problem through DSC maximization using GA, the chromosomes representation, selection, crossover and mutation operators in addition to an operator called scattering is applied on the offspring. Initially, each sensor randomly joins a group among prescribed groups. Then a group forms a cover if it can cover all targets. They considered that it is impossible to find more than z disjoint covers if z sensors only cover a target as a required minimum number of sensors $S(T)$. Thus, the number of the prescribed groups is used as an upper bound (ub) of the number of covers. The fitness of a chromosome is defined as the number of disjoint covers that can be found by the grouping combination represented by the chromosome. The uniform crossover exchanges each gene of the two parents and the mutation exchanges a randomly selected gene to a random value from 1 to ub to the new generation. We programmed this existing DSC based method and applied it in parallel with our GA-based scheduling method incorporating NDSC GA, to improve the lifetime for the same instances. Regarding the number of covers generated by the DSCGA and

the NDSCGA, the table 4.11 synthesizes this comparison for different numbers of sensors.

TABLE 4.11: MSCGA and NDSCGA

Num. of Sensors	100	200	300	400	500
Num ofNDSC	3924	5443	6650	7781	9013
DSC-upper bound	25	52	64	84	107
Num of DSC	23	46	73	77	99

It is clear that the number of NDSC is greater than DSC, but additional effort is required to schedule this larger number for a lifetime greater than or equal to the lifetime obtained by the DSC method. Therefore, the maximum number of NDSC generated by NDSCGA is optimally scheduled using the genetic algorithm based scheduling method to reach the optimal wireless sensor network lifetime. The four strategies of GA explained and the DSCGA are applied to the same instances as in figure 4.19. It is clear that our NDSCGA gives better solutions compared to DSCGA for the same instances.

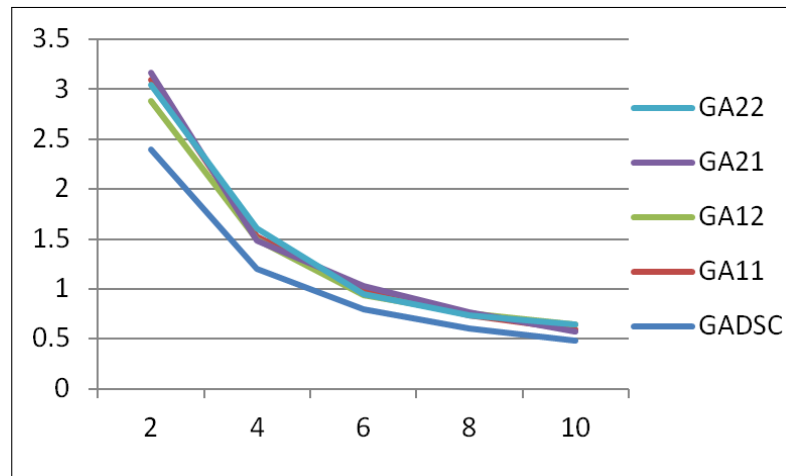


FIGURE 4.19: GA with NDSC via GA with DSC.

The DSC based method is applied with our genetic algorithm to calculate the lifetime for the same instances. For the NDSCGA, the average of all the strategies GA11, GA12, GA21, and GA22 is used to be compared to the result obtained by DSCGA as in table 4.12.

TABLE 4.12: DSCGA and NDSCGA

Sensors	E_i	$E_i(k)$	MSCGA	NDSCGA	percentage
10	160	2	240	298	24.17
10	160	4	120	146	21.67
10	160	6	80	98	22.5
10	160	8	60	75	25
10	160	10	48	61	27.08
Average					24.08

The NDSCs based GA brought out an average enhancement about 24.08 percent compared to the GA based on DSCs.

4.8 Conclusion

In this chapter, we have described the results obtained for the proposed methods used with WSNs lifetime optimization. We have investigated the proposed methods for this problem specified using NDSCs and solved in two subproblems: 1) the NDSCs finding subproblem and 2) the NDSCs scheduling and lifetime optimization subproblem. For the first subproblem, two methods are investigated: the binary representation based method and the GA based method. The GA based method has successfully avoided the limitation of the binary representation based method in term of scalability. The investigation has proven that the number of sensors could be expanded, which gives the method we developed the capability to be used in scalable surveillance systems. The NDSCs covers generated in subproblem one are implemented in subproblem two to be scheduled for WSNs lifetime optimization. The optimal solution for this problem is found using ILP mathematical model and the Cplex software for the exact solution and GA based method for faster solution. The investigation has proven that one can find high-quality solutions compared to the exact solutions. The different mutation and

crossover operators have been implemented to investigate and evaluate these GA-based strategies on various instances. The integration of the two subproblems has provided the capability to use the mathematical model provided in this work to find the optimal solution for this problem specified with NDSC. The investigation has proved that the NDSC based method can find better solutions compared to an existing DSC based method. This work has provided one additional efficient method for WSNS lifetime optimization problem with more advantages. One of these advantages is the possibility for the different types of sensing nodes with different characteristics, operational energy behavior, and functions, to be efficiently used together in the same WSNs based system.

General conclusion and perspectives

Our work investigates the scheduling and optimization techniques applicable to optimize the WSNs lifespan under energy constraints. In the existing literature, this problem has been formulated as disjoint sets cover (DSCs) to maximizing the network lifetime under limited energy reserve. Among the works in the literature, the linear programming, genetic algorithms, and other optimization methods have been used to maximize WSN lifetime based on DSC. The investigations have shown that the valid amount of energy was not used because of the DSC constraints. In our work, we consider the problem of WSNs lifespan optimization as scheduling of non-disjoint sets covers NDSCs. We have formulated this problem, using the integer linear programming (ILP) mathematical model, then we developed a method based on genetic algorithm GA to find the optimal lifespan. First, we used the Cplex to find the exact solution for small size instances, while the sensors have initial energy reserve and energy consumption rate. The results we obtained explained that the NDSCs is promising compared to the DSCs based methods.

Then, as a contribution, due to the complexity of the problem, we have developed a GA-based scheduling method to solve efficiently this problem to extending the lifetime, under different operation conditions, while considering sensors with different initial energy reserve and energy consumption rate. Also, we proposed heuristics based on GA considering different crossover and mutation strategies. The algorithm of the proposed method has been coded and simulated using C language programming. One can find the results in the Proceedings of IEEE IDAACS'2015

international conference on Advanced Data Acquisition and Intelligent Data Processing.

We have also developed an integer linear programming model for scheduling and maximizing the WSN lifespan successfully for NDSCs. For that purpose, we considered a set S of sensors, randomly deployed to monitor a set T of targets. We determined the set of q NDSCs, and we proposed a method that aims at finding the optimal scheduling for extending the WSN lifetime. The method has been applied using eclipse-supported program with the linear optimization libraries of Cplex and Java Development Kit (JDK) to obtain the optimal solution for this problem on different operating conditions through numerical experiments. The results were presented in the proceedings of CIE45 international conference on Computers and Industrial Engineering.

Also, we proposed an exact method and GA-based heuristics, aiming at maximizing WSNs lifespan through scheduling, with a comparative study of both methods regarding execution time and quality of solution for different instances. Then, we developed a new GA-based method that can find the optimal number of NDSCs to be scheduled using the previous method.

The DSC from the literature and the last NDSC-based methods were programmed for evaluation. Our NDSCs based method used for solving WSNs lifetime optimization problems under constraints of limited reserve of energy and has provided promising results that were presented at the 3rd IEEE IDAACS's SWS'2016 Symposium on Wireless Systems.

Further work on the GA-based method involving new crossover and mutation operators have brought out better performances. The results were included in the proceedings of the 14th IFAC PDeS'2016 International Conference on Programmable Devices and Embedded Systems. As perspectives we would propose further investigations regarding the followings:

- WSNs lifetime modeling: Several factors affect the WSNs lifetime beside its energy constraints, such as failure hazard, component performance degradation through time and environmental influences. The model proposed in this

work could be extended to include further realistic influent factors to bring out an integrated vision of WSNs lifetime.

- Study of WSNs reliability and resiliency: this study would rely on the ND-SCs' properties offering the possibility to a sensor to participate in one or more covers. The results will then be utilized for repairing and reconfiguring covers in case of some sensors failure and thus enhance the reliability and resilience of the WSNs.
- Scheduling: The model and method proposed in this work could extend to solving scheduling problems with resources availability constraints.
- Optimization: Lastly, a further works could focus on reducing the complexity of the proposed exact method to deal with larger size instances.

Conclusion générale

Notre travail est consacré à l'étude des techniques d'optimisation de la durée de vie des réseaux de capteurs sans fil (RCSF) sous contrainte d'énergie. Ce problème a été formulé dans la littérature existante sous forme d'ordonnancement d'ensembles de couvertures disjoints (DSC) pour maximiser la durée de vie du réseau avec une réserve limitée d'énergie. Parmi les travaux en cours, la programmation linéaire, les algorithmes génétiques et d'autres méthodes d'optimisation ont été utilisées pour maximiser la durée de vie WSN à l'aide de DSCs. Les recherches ont montré que toute la quantité d'énergie disponible n'est pas utilisée au maximum en raison des contraintes des DSC. Dans notre travail, nous considérons le problème d'optimisation de la durée de vie des RCSFs comme un problème d'ordonnancement des ensembles couvertures non-disjoints (ECND). Nous avons utilisé un modèle mathématique de programmation linéaire en nombres entiers (ILP) pour résoudre ce problème, puis, nous avons développé une méthode basée sur les algorithmes génétiques (GA) pour obtenir une durée de vie optimale du réseau. D'abord, nous avons utilisé le Cplex pour trouver une solution exacte pour les instances de petite taille, avec des capteurs ayant les mêmes réserves initiales d'énergie et le même taux de consommation d'énergie. Les résultats obtenus ont montré que le NDSC est prometteur par rapport aux méthodes basées les DSCs. Ensuite, en raison de la complexité du problème, nous avons développé une méthode d'ordonnancement basée les GA pour résoudre efficacement le problème d'extension de la durée de vie, sous différentes conditions de fonctionnement, tout en tenant compte des capteurs avec différentes réserves initiales et taux de consommation d'énergie. En outre, nous avons proposé des heuristiques basées sur les GA suivant

différentes stratégies de croisement et de mutation. L'algorithme de la méthode proposée a été codé, programmé et simulé en utilisant le langage C. Les résultats de cette contribution sont accessibles dans les actes de la conférence internationale IDAACS'2015 (Advanced Data Acquisition and Intelligent Data Processing).

Nous avons également développé avec succès un modèle d'ordonnancement et de maximisation de la durée de vie de RCSFs par programmation linéaire pour NDSCs. Pour cela, nous avons considéré un ensemble S de capteurs, déployés au hasard pour suivre un ensemble T de cibles. Nous avons pu déterminer l'ensemble des q NDSCs, et nous avons proposé une méthode visant à déterminer l'ordonnancement optimal permettant de prolonger la durée de vie de WSN. La méthode a été appliquée en utilisant le programme dans un environnement Eclipse soutenu par des bibliothèques d'optimisation linéaires de Cplex et Java Development Kit (JDK) pour obtenir la solution optimale à ce problème dans différentes conditions de fonctionnement grâce à des expérimentations numériques. Les résultats sont publiés dans les actes de la conférence internationale CIE45 (Computers and Industrial Engineering). En outre, nous avons proposé une méthode exacte et des heuristiques à base des GA, visant à maximiser la durée de vie des RCSFs par ordonnancement, avec une étude comparative des deux méthodes développées, en termes de temps d'exécution et de qualité de solution pour les différentes instances. Ensuite, nous avons développé une nouvelle méthodes basées les GA permettant trouver le nombre optimal de NDSCs à ordonnancer à l'aide de la méthode précédente. Aussi bien les méthodes basées sur les DSCs dans la littérature que les nôtres basées sur les NDSCs ont été programmées et évaluées. Notre méthode, basée sur les NDSCs, a été appliquée à la résolution des problèmes d'optimisation de la durée de vie des RCSFs sous contraintes de réserve limitée d'énergie avec succès. Elle a fourni des résultats prometteurs, exposés dans les actes du 3ème symposium IEEE IDAACS SWS'2016 (Symposium on Wireless Systems). Des travaux complémentaires sur la méthode basée sur GA impliquant de nouveaux opérateurs de croisement et de mutation ont mis en évidence de meilleures performances. Les résultats ont été inclus dans les actes de la 14ème Conférence internationale PDeS'2016 IFAC (Programmable Devices and Embedded Systems).

Comme perspectives, nous proposerons d'autres travaux sur les points suivants:

- Modélisation de la durée de vie du RCSFs: Plusieurs facteurs affectent la durée de vie des RCSFs en dehors des contraintes d'énergie, tels que le risque de défaillance, la dégradation des composants et les influences de l'environnement. Le modèle proposé dans ce travail pourrait être étendu afin d'inclure d'autres facteurs d'influence plus réalistes pour faire ressortir une vision intégrée de la durée de vie des RCSFs.
- Fiabilité et résilience des RCSFs: Cette étude s'appuiera sur les propriétés des NDSC offrant la possibilité, à un capteur de participer à une ou plusieurs couvertures. Les résultats seront ensuite utilisés pour la réparation et la reconfiguration des couvertures en cas de défaillance de certains capteurs, et améliorer ainsi la fiabilité et la résilience des RCSFs.
- Planification: Le modèle et la méthode proposés dans ce travail pourraient s'étendre aux problèmes d'ordonnancement, avec des contraintes de disponibilité des ressources.
- Optimisation: Enfin, d'autres travaux futurs pourraient s'intéresser réduire la complexité de la méthode exacte proposée afin de traiter les instances de plus grande taille.

Bibliography

- [1] G. J. Pottie, “Wireless sensor networks,” in *Proc. of the IEEE Information Theory Workshop*, pp. 139–140, 1998.
- [2] G. Xu, W. Shen, and X. Wang, “Applications of wireless sensor networks in marine environment monitoring: A survey,” *Sensors*, vol. 14, pp. 16932–16954, 2014.
- [3] Q. Wang and I. Balasingham, *Wireless Sensor Networks - An Introduction*, ch. 1. In Tech, 2010.
- [4] M. Won, C. Yang, W. Zhou, and R. Stoleru, “Energy efficient multi-channel media access control for dense wireless ad hoc and sensor networks,” *Wireless Networks*, vol. 19, pp. 1537–1551, 2013.
- [5] L. M. Borges, S. Member, F. J. Velez, S. Member, and A. S. Lebres, “Survey on the characterization and classification of wireless sensor network applications,” *IEEE Communication Surveys and Tutorials*, vol. 16, no. 4, pp. 1860 – 1890, 2014.
- [6] O. Vermesan and P. Friess, *Internet of things from research and innovation to market and deployment*. River Publishers, 2014.
- [7] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, “Survey paper-internet of things: Vision, applications and research challenges,” *Ad Hoc Networks*, vol. 2012, pp. 1497–1516, 2012.

- [8] N. Nidal and Y. Chen, "Secure and energy-efficient multipath routing protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 11, pp. 2401–2412, 2007.
- [9] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor network: A survey," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 707–718, 2002.
- [10] M. E. Keskin, I. K. Altinel, N. Aras, and C. Ersoy, "Wireless sensor network lifetime maximization by optimal sensor deployment, activity scheduling, data routing and sink mobility," *Ad Hoc Networks*, vol. 17, pp. 18–36, 2014.
- [11] A. Dhawan, *Maximum Lifetime Scheduling in Wireless Sensor Networks*, ch. 2. InTech, 2012.
- [12] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [13] S. S. Rao, *Engineering Optimization: Theory and Practice*. John Wiley, 4th edition ed., 2009.
- [14] M. Mitchell, *An Introduction to Genetic algorithms*. MIT Press, 5th printing ed., 1998.
- [15] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks*. John Wiley, 2010.
- [16] J.-H. Kim, D.-H. Kim, H.-Y. Kwak, and Y.-C. Byun, "Integration between WSNs and internet based on address internetworking for Web service, computing and informatics," *Computing and Informatics*, vol. 27, pp. 707–718, 2008.
- [17] S. P. Singh and S. C. Sharma, "A survey on research issues in wireless sensor networks," *Open Transactions on Wireless Sensor Network*, vol. 2, no. 1, pp. 1–18, 2015.

- [18] K. Kaushal, T. Kaur, and J. Kaur, "Zigbee based wireless sensor networks," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, pp. 7752–7755, 2012.
- [19] N. A. Somani and Y. Patel, "Zigbee: A low power wireless technology for industrial applications," *International Journal of Control Theory and Computer Modelling*, vol. 2, no. 3, pp. 27–33, 2012.
- [20] B. A. Mahmood and D. Manivannan, "Position based and hybrid routing protocols for mobile ad hoc networks: A survey," *Wireless Personal Communication*, vol. 83, pp. 1009–1033, 2015.
- [21] M. M. Qabajeh, A. H. Abdalla, O. O. Khalifa, and L. K. Qabajeh, "A survey on scalable multicasting in mobile ad hoc networks," *Wireless Personal Communication*, vol. 80, pp. 369–393, 2014.
- [22] C. F. García-Hernández, P. H. Ibargüengoytia-González, J. García-Hernández, and J. A. Pérez-Díaz., "Wireless sensor networks and applications: a survey," *International Journal of Computer Science and Network Security*, vol. 7, no. 3, pp. 264–273, 2007.
- [23] K. K. Khedo, R. Perseedoss, and A. Mungur, "Wireless sensor networks air pollution monitoring system," *International Journal of Wireless and Mobile Networks*, vol. 2, no. 2, 2010.
- [24] D. Ye, D. Gong, and W. Wang, "Application of wireless sensor networks in environmental monitoring," in *Proc. of the IEEE 2nd Int. Conf. on Power Electronics and Intelligent Transportation System*, vol. 1, pp. 205–208, 2009.
- [25] M. F. Othmana and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," *Intl. Sym. on Robotics and Intelligent Sensors*, vol. 2012, no. 41, pp. 1204–1210, 2012.
- [26] M. Paavola and K. Leiviskä, *Wireless Sensor Networks in Industrial Automation*, ch. 10. In Tech, 2010.

- [27] M. P. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic, "A survey of military applications of wireless sensor networks," in *Proc. of the IEEE Mediterranean Conf. on Embedded Computing (MECO'2012)*, pp. 196–199, 2012.
- [28] S. JA, C. Q, D. T, F. L, H. Z, K. R, L. S, S. S, S. R, and W. A, "Wireless sensor networks for in-home healthcare: Potential and challenges," in *Proc. of the Workshop on High confidence medical device software and systems (HCMDSS'2005)*, pp. 2–3, 2005.
- [29] M. A. Kafi, Y. Challal, D. Djenouri, M. Doudou, A. Bouabdallah, and N. Badache, "A study of wireless sensor networks for urban traffic monitoring: Applications and architectures," *Procedia Computer Science*, vol. 19, pp. 617–626, 2013.
- [30] B. Bagheri, S. Yang, H.-A. Kao, and J. Lee, "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1622–1627, 2015.
- [31] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.
- [32] N. Srivastava, "Challenges of next-generation wireless sensor networks and its impact on society," *Journal of Telecommunications*, vol. 1, no. 1, pp. 128–133, 2010.
- [33] M. C. Akewar and N. V. Thakur, "A study of wireless mobile sensor network deployment," *International Journal of Computer Networks and Wireless Communications*, vol. 2, no. 4, pp. 533–541, 2012.
- [34] K. Xu, Q. Wang, H. Hassanein, and G. Takahara, "Optimal wireless sensor networks WSNs deployment: Minimum cost with lifetime constraint," *Proc. of the IEEE Int. Conf. on Wireless And Mobile Computing, Networking And Communications (WiMob'2005)*, vol. 3, pp. 454 –461, 2005.

- [35] A.-J. Garcia-Sanchez, F. Garcia-Sanchez, F. Losilla, P. Kulakowski, J. Garcia-Haro, A. Rodríguez, J.-V. López-Bao, and F. Palomares, “Wireless sensor network deployment for monitoring wildlife passages,” *Journal of Sensors*, vol. 10, pp. 7236–7262, 2010.
- [36] A. K. Idrees, K. Deschinkel, M. Salomon, and R. Couturier, “Distributed lifetime coverage optimization protocol in wireless sensor networks,” *The Journal of Supercomputing*, vol. 71, no. 12, pp. 4578–4593, 2015.
- [37] M. C. Akewar and N. V. Thakur, “Grid based wireless mobile sensor network deployment with obstacle adaptability,” *International Journal of Wireless and Mobile Networks*, vol. 4, no. 5, p. 21, 2012.
- [38] T. H. Lim, I. Bate, and J. Timmis, “A self-adaptive fault-tolerant systems for a dependable wireless sensor networks,” *Springer journal of Design Automation for Embedded Systems*, vol. 18, no. 3-4, pp. 223–250, 2014.
- [39] Y. Yun and Y. Xia, “A method for deciding node densities in non-uniform deployment of wireless sensors,” in *Proc. of the 11th Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt’2013)*, pp. 264–271, 2013.
- [40] Y. Huang, J.-F. Martínez, J. Sendra, and L. López, “The influence of communication range on connectivity for resilient wireless sensor networks using a probabilistic approach,” *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–11, 2013.
- [41] J. A. Khan, H. K. Qureshi, and A. Iqbal, “Energy management in wireless sensor networks: A survey,” *Computers and Electrical Engineering*, vol. 41, pp. 159–176, 2014.
- [42] J. Jia, J. Chena, G. Changa, and Z. Tana, “Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm,” *Computers and Mathematics with Applications*, vol. 57, pp. 1756–1766, 2009.

- [43] Z. Sun, W. Wu, H. Wang, H. Chen, and W. Wei, "An optimized strategy coverage control algorithm for WSN," *International Journal of Distributed Sensor Networks*, vol. 2014, pp. 1–12, 2014.
- [44] Z. Jiucui, C. Song, S. Hamid, and A. Mahmoud, "Lifetime optimization for wireless sensor networks using the nonlinear battery current effect," in *Proc. of the IEEE Int. Conf. on Communications*, pp. 1–6, 2009.
- [45] T. K. Jain, D. S. Saini, and S. V. Bhooshan, "Lifetime optimization of a multiple sink wireless sensor network through energy balancing," *Journal of Sensors*, 2015.
- [46] H. Wang, N. Agoulmine, M. Ma, and Y. Jin, "Network lifetime optimization in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1127–1137, 2010.
- [47] A. Nashat, Y.-A. Le Borgne, K. Steenhaut, and A. Nowé, "Lifetime optimization for wireless sensor networks with correlated data gathering," in *Proc. of the IEEE 8th Int. Symp. on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (WiOpt'2010)*, pp. 266–272, 2010.
- [48] R. Asorey-Cacheda, A. J. García-Sánchez, F. García-Sánchez, J. García-Haro, and F. J. González-Castano, "On maximizing the lifetime of wireless sensor networks by optimally assigning energy supplies," *Journal of Sensors*, vol. 13, no. 8, pp. 10219–10244, 2013.
- [49] A. Dâmaso, N. Rosa, and P. Maciel, "Reliability of wireless sensor networks," *Journal of Sensors*, vol. 14, pp. 15760–15785, 2014.
- [50] D. He, G. Mujica, J. Portilla, and T. Riesgo, "Modelling and planning reliable wireless sensor networks based on multi-objective optimization genetic algorithm with changeable length," *Journal of Heuristics*, vol. 5, no. 21, pp. 257–300, 2015.

- [51] L. Venkatesan, S. Shanmugavel, and C. Subramaniam, "A survey on modeling and enhancing reliability of wireless sensor network," *Journal of Wireless Sensor Network*, vol. 5, pp. 41–51, 2013.
- [52] F. D. Macias-Escriba, R. Haber, R. del Toro, and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and application," *Expert Systems with Applications*, vol. 5, no. 40, pp. 7267–7279, 2013.
- [53] P. M. L, *Planning and scheduling in manufacturing and services*, vol. 24. Springer, 2005.
- [54] I. Kacem, "Effective algorithms for scheduling problems under non-availability constraints," in *Proc. of the IEEE 2nd Int. Conf. on Systems and Computer Science (ICSCS'2013)*, pp. 162–169, 2013.
- [55] P. M. L, *Scheduling: theory, algorithms and systems*. Springer, 2011.
- [56] A. Rossi, M. Sevaux, A. Singh, and M. J. Geiger, "The cover scheduling problem arising in wireless sensor networks," Research Report RR-11-02-01, Helmut-Schmidt-Universität, Institut für betriebliche Logistik und Organisation, Hamburg, Germany, 2011.
- [57] A. Rossi, A. Singh, and M. Sevaux, "Génération de colonnes et réseaux de capteurs sans fil," In *Proc. of the ROADEF 11ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, 2010.
- [58] T. Yuan, E. Eylem, and O. Fusun, "Energy-constrained task mapping and scheduling in wireless sensor networks," in *Proc. of the IEEE Int. Conf. on Mobile Adhoc and Sensor Systems*, pp. 211–218, 2005.
- [59] I. Kacem, "Approximation algorithms for the makespan minimization with positive tails on a single machine with a fixed non-availability interval," *Journal of Combinatorial Optimization*, vol. 17, pp. 117–133, 2009.

- [60] A. Levin and M. Penn, “Approximation algorithm for minimizing total latency in machine scheduling with deliveries,” *Journal of Discrete Optimization*, vol. 5, pp. 97–107, 2008.
- [61] Q. Zhen, S. Cliff, and Z. Yuan, “Minimizing the total weighted completion time of coflows in datacenter networks,” in *Proc. of the ACM 27th Symposium on Parallelism in Algorithms and Architectures*, pp. 294–303, 2015.
- [62] A. G., C. P., G. G., K. V., M.-S. A., and P. M., *Complexity and approximation combinatorial optimization problems and their approximability properties*. Springer, 2012.
- [63] Franz, *Optimization Problems*, ch. 1. Springer, 2011.
- [64] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms, Combinatorial Optimization*. Springer, 2010.
- [65] S. S. Rao, *Engineering optimization theory and practice*. John Wiley, 4th edition ed., 2009.
- [66] S. Soliman and A. Mantawy, *Mathematical Optimization Techniques*, ch. 2. Springer, 2012.
- [67] R. Marti and G. Reinelt, *The Linear Ordering Problem Exact and Heuristic Methods in Combinatorial Optimization*, vol. 175. Springer Science & Business Media, 2011.
- [68] M. Yang and J. Liu, “A maximum lifetime coverage algorithm based on linear programming,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 2, pp. 296–301, 2014.
- [69] B. Korte and J. Vygen, *Combinatorial Optimization Theory and Algorithms*. Springer, 2nd ed. ed., 2002.
- [70] Y.-h. Kim, H.-J. Lee, Y.-H. Han, and Y.-S. Jeong, “A branch and bound algorithm for extending the lifetime of wireless sensor networks,” in *Proc. of the IEEE 70th Vehicular Technology Conference Fall (VTC’2009-Fall)*, pp. 1–5, 2009.

-
- [71] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Springer, 2007.
- [72] M. Dianati, I. Song, and M. Treiber, “An introduction to genetic algorithms and evolution strategies,” technical report, University of Waterloo, Electrical and Computer Engineering, 200 Univ. Ave. West, Ontario, N2L 3G1, Canada, 2003.
- [73] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, 1999.
- [74] M. R. and R. G., *Heuristic Methods*, vol. 175, ch. 2. Springer, 2011.
- [75] C. Sevgi and A. Koçyigit, “Optimal deployment in randomly deployed heterogeneous WSNs a connected coverage approach,” *Journal of Network and Computer Applications*, vol. 46, no. 2014, 2014.
- [76] R. J. Barton and R. Zheng, “Order-optimal data aggregation in regular wireless sensor networks,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5811–5821, 2010.
- [77] Y. Chen, Z. Wang, T. Ren, and H. Lv, “Lifetime optimization algorithm with mobile sink nodes for wireless sensor networks based on location information,” *International Journal of Distributed Sensor Networks*, vol. 2015, p. 16, 2015.
- [78] S. M and J. S. Hephzibah, “Survey on improved lifetime of wsn using optimized scheduling,” *Karpagam Journal of Engineering Research*, vol. 1, no. 1, pp. 1–4, 2014.
- [79] M. Cardei and D. Z. Du, “Improving wireless sensor network lifetime through power aware organization,” *Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.
- [80] N. Ahn and S. Park, “A new mathematical formulation and a heuristic for the maximum disjoint set covers problem to improve the lifetime of the

- wireless sensor network,” *Ad Hoc and Sensor Wireless Networks*, vol. 13, pp. 209–225, 2011.
- [81] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” *IEEE Int. Conf. on Communications, Helsinki, Finland*, vol. 2, pp. 472–476, 2001.
- [82] R. Cerulli, R. D. Donato, and A. Raicon, “Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges,” *European Journal of Operational Research*, vol. 220, no. 1, pp. 58–66, 2012.
- [83] C.-C. Lai, C.-K. Ting, and R.-S. Ko, “An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications,” in *Proc. of the IEEE CEC’2007 Congress on Evolutionary Computation*, vol. 2, (Singapore), pp. 3531–3538, 2007.
- [84] M. K. Singh and P. Nagarathna, “Improvement in life span of wireless sensor network using genetic algorithm with new fitness function,” *IEEE Int. Conf. on Electronics Computer Technology, Kanyakumari, India*, vol. 2, pp. 1–23, 2012.
- [85] V. K. Bagaria, A. Pananjady, and R. Vaze, “Optimally approximating the lifetime of wireless sensor networks,” *arXiv preprint*, vol. 2, pp. 3531–3538, 2013.
- [86] S. Ropke, “Heuristic and exact algorithms for vehicle routing problems,” *Unpublished PhD thesis, Computer Science Department, University of Copenhagen*, 2005.
- [87] H. Liu, X. Jia, P.-J. Wan, C.-W. Yi, S. K. Makki, and N. Pissinou, “Maximizing lifetime of sensor surveillance systems,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 334–345, 2007.

- [88] R. Montemanni, “Integer programming formulations for maximum lifetime broadcasting problems in wireless sensor networks,” *Wireless Sensor Network*, vol. 2, pp. 924–935, 2010.
- [89] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in *Proc. of the IEEE 24th Annual Joint Conference of the Computer and Communications Societies*, vol. 3, pp. 1976–1984, 2005.
- [90] M. Cardei, J. Wu, M. Lu, and M. O. Pervaiz, “Maximum network lifetime in wireless sensor networks with adjustable sensing ranges,” in *Proc. of the IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications*, vol. 3, pp. 438–445, 2005.
- [91] A. Singh and A. Rossi, “A genetic algorithm based exact approach for lifetime maximization of directional sensor networks,” *Ad Hoc Networks*, vol. 11, pp. 1006–1021, 2013.
- [92] K. H. Adjallah, N. V. Lozano, and H. Snoussi, “Sensors network management for health condition data coverage on large-scale infrastructures,” *IFAC Proceedings Volumes*, vol. 43, no. 3, pp. 116–121, 2010.
- [93] D. Dame and T. Ousmane, “Alarm: An energy aware sleep scheduling algorithm for lifetime maximization in wireless sensor networks,” in *Proc. of the IEEE Symposium on Wireless Technology & Applications (ISWTA’2013)*, pp. 74–79, 2013.
- [94] J. Carle and D. Simplot-Ryl, “Energy-efficient area monitoring for sensor network,” *IEEE Computer*, vol. 37, no. 2, pp. 40–46, 2004.
- [95] C. Srilakshmi, “A genetic replacement algorithm for optimizing wireless sensor networks,” in *Proc. of the Int. Conf. on Science Engineering and Management Research (ICSEMR’2014)*, pp. 1–6, 2014.

- [96] Y.-H. Han, C.-M. Kim, and J.-M. Gil, "A greedy algorithm for target coverage scheduling in directional sensor networks," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 1, pp. 96–106, 2010.
- [97] H. Mohamadi, S. Salleh, and M. N. Razali, "Heuristic methods to maximize network life time in directional sensor networks with adjustable sensing ranges," *Journal of Network and Computer Applications*, vol. 46, no. 2, pp. 26–35, 2014.
- [98] K. Deschinkel, "A column generation based heuristic for maximum lifetime coverage in wireless sensor networks," in *Proc. of the 5th Int. Conf. on Sensor Technologies and Applications (SENSORCOMM'11)*, pp. 209–214, 2015.
- [99] F. Castaño, A. Rossi, M. Sevaux, and N. Velasco, "A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints," *Computers and Operations Research*, vol. 52, pp. 220–230, 2013.
- [100] D. Zorbas, D. Glynos, P. Kotzanikolaou, and C. Douligeris, "B {GOP}: An adaptive algorithm for coverage problems in wireless sensor networks," in *Proc. of the 13th European Wireless Conference*, 2007.
- [101] D. Zorbas, D. Glynos, P. Kotzanikolaou, and C. Douligeris, "Solving coverage problems in wireless sensor networks using cover sets," *Ad Hoc Networks*, vol. 8, pp. 400–415, 2010.
- [102] H. Mohamadi, A. S. Ismail, and S. Salleh, "Solving target coverage problem using cover sets in wireless sensor networks based on learning automata," *Wireless Personal Communications*, vol. 75, pp. 447–463, 2014.
- [103] K. Genova and V. Guliashk, "Linear integer programming methods and approaches : A survey," *Bulgarian Academy of Sciences, Cybernetics and Information Technologies*, vol. 11, pp. 447–463, 2011.