



HAL
open science

Decomposition-based approaches for the design of energy efficient wireless sensor networks

Fabian Castaño

► **To cite this version:**

Fabian Castaño. Decomposition-based approaches for the design of energy efficient wireless sensor networks. Operations Research [math.OC]. Université Européenne de Bretagne; Université de Bretagne-Sud; Universidad de Los Andes, 2014. English. NNT: . tel-01096440

HAL Id: tel-01096440

<https://hal.science/tel-01096440>

Submitted on 17 Dec 2014

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.



 **Universidad de
los Andes**
Facultad de Ingeniería



THÈSE
UNIVERSITÉ DE BRETAGNE-SUD
UNIVERSIDAD DE LOS ANDES

présentée par

Fabián Andrés Castaño Giraldo

sous le sceau de l'Université européenne de Bretagne
pour obtenir le titre de
DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE-SUD
DOCTOR DE LA UNIVERSIDAD DE LOS ANDES
Mention: STIC
École doctorale SICMA

Decomposition-based approaches for the design of energy efficient wireless sensor networks

Thèse soutenue le 1er Octobre 2014,
devant la commission d'examen composée de :

M. Raffaele Cerulli
Professeur des Universités, Università degli studi di Salerno / Président

M. Marco Lübbecke
Professeur des Universités, RWTH Aachen University / Rapporteur

M. Patrick Siarry
Professeur des Universités, Université Paris-Est Créteil Val-de-Marne / Rapporteur

M. Ciro Amaya
Professeur des Universités, Universidad de los Andes / Examineur

M. Faïcel Hnaïen
Maître de Conférences, Université de Technologie de Troyes / Examineur

M. André Rossi
Maître de Conférences HDR, Université de Bretagne-Sud / Co-directeur de thèse

Mme. Nubia Velasco
Professeur des Universités, Universidad de los Andes / Directrice de thèse

M. Marc Sevaux
Professeur des Universités, Université de Bretagne-Sud / Directeur de thèse

Résumé

La gestion de l'énergie est une préoccupation majeure dans les réseaux de capteurs sans fil. Ces capteurs sont généralement alimentés par une batterie embarquant une quantité d'énergie finie. Par conséquent, le temps pendant lequel les capteurs peuvent surveiller une zone et communiquer par signaux radio est limité lorsqu'il n'est pas possible de remplacer leur batterie. En outre, les réseaux de capteurs sont parfois déployés dans des zones difficiles d'accès ou dans des environnements hostiles dans lesquels le placement des capteurs peut être considéré comme aléatoire (c'est le cas par exemple lorsque les capteurs sont largués d'un avion ou d'un hélicoptère). Ainsi, l'emplacement des capteurs n'est pas connu a priori et les approches pour utiliser efficacement l'énergie sont nécessaires. Cette thèse explore l'utilisation de la génération de colonnes pour optimiser l'utilisation de l'énergie dans les réseaux de capteurs sans fil. La génération de colonnes peut être vue comme un cadre général pour résoudre différents problèmes dans la conception et l'exploitation de ces réseaux. Plusieurs versions du problème et divers modèles sont proposés pour représenter leur fonctionnement, en utilisant notamment la génération de colonnes. Ces approches exploitent le caractère naturel de la génération de colonnes pour modéliser les différents aspects des réseaux de capteurs sans fil. Dans cette thèse, des contributions algorithmiques sont apportées afin de tirer le meilleur parti de la génération de colonnes au plan de l'efficacité computationnelle. Des stratégies hybrides combinant génération de colonnes et (méta)-heuristiques et donnant lieu à des méthodes exactes et approchées sont proposées et évaluées. Des tests numériques montrent l'efficacité des approches proposées et procurent des bornes supérieures qui peuvent être employées pour évaluer l'efficacité des méthodes centralisées et distribuées. Enfin, des perspectives sont dégagées concernant les performances et la portabilité de la génération de colonnes pour aborder des problèmes plus réalistes et tenir compte des caractéristiques des réseaux de capteurs sans fil du futur.

Mots clés: Réseau de capteurs sans fil, Durée de vie, Génération de colonnes, Contraintes de connectivité

Resumen

En el diseño de redes inalámbricas una de las principales preocupaciones es cómo hacer un uso eficiente de la energía. Estas redes están compuestas de sensores que son alimentados por una fuente externa de energía (batería) que tiene una capacidad limitada y, en muchos casos es irremplazable. Este hecho limita el tiempo durante el cual los sensores pueden realizar su trabajo de monitoreo a una zona y transmitir las señales percibidas. Por otro lado, en algunas aplicaciones se requiere que los sensores estén ubicados en lugares de difícil acceso u hostiles, impidiendo su ubicación manual, lo que implica que el posicionamiento es aleatorio (p.ej. lanzados desde un helicóptero, un "dron" o un avión). Dadas todas estas circunstancias, el uso de estrategias para optimizar el uso de energía es requerido. En esta tesis se explora el uso de modelos basados en redes para explicar el consumo de energía en una red de sensores inalámbricos y la técnica de generación de columnas como herramienta de solución para la optimización del uso de dicha energía. La técnica de generación de columnas ofrece una herramienta de solución que puede usarse para representar con naturalidad la asignación y programación de la operación de los sensores. En este trabajo se proponen diferentes mejoras que permiten tomar ventaja de la técnica de generación de columnas y mantener su eficiencia. Algunas estrategias híbridas que combinan el principio de la descomposición con metaheurísticas y métodos exactos son propuestas. Las propuestas fueron extensamente evaluadas y los experimentos computacionales confirman la eficacia de los métodos, proporcionando cotas superiores que pueden ser usados para evaluar el rendimiento de otros métodos centralizados y distribuidos. Finalmente, algunas direcciones de investigación son dilucidadas sobre la base de este método y su uso como estrategia para atacar problemas más realistas que consideren las características de las redes de sensores inalámbricos del futuro.

Palabras clave: Redes sensores inalámbricos, Vida útil, Generación de Columnas, Restricciones de conectividad

Abstract

Energy is a major concern in wireless sensor networks (WSN). These devices are typically battery operated and provided with a limited amount of energy. As a consequence, the time during which sensors can monitor the interesting phenomena and communicate through wireless signals might be limited because of (sometimes) irreplaceable batteries. Additionally, it is very common for WSN to be used in remote or hostile environments which possibly makes necessary a random placement strategy (by using an airplane, a drone or a helicopter). Hence, the sensors location is not known a priori and approaches to efficiently use the energy are needed to answer to network topologies only known after sensors deployment. This thesis explores the use of column generation to efficiently use the energy in WSN. It is shown that column generation can be used as a general framework to tackle different problems in WSN design. Several versions of the problem and models for the operation of the WSN are adapted to be solved through column generation. These approaches take advantage of the natural way that column generation offers to consider different features of the WSN operation. Additionally, some computational improvements are proposed to keep the column generation method operating as an efficient exact approach. Hybrid strategies combining column generation with (meta)heuristic and exact approaches are considered and evaluated. The computational experiments demonstrate the efficiency of the proposed approaches and provide practitioners on WSN research with strategies to compute upper bounds to evaluate heuristic centralized and decentralized approaches. Finally, some future directions of research are provided based on the performance and adaptability of column generation to consider more sophisticated models and characteristics newly introduced in sensor devices.

Keywords: Wireless sensor networks, Lifetime, Column generation, Connectivity constraints



n d'ordre :

Université de Bretagne-Sud

Centre de Recherche Christiaan Huygens - BP 92116, 56321 Lorient CEDEX
Tél : + 33(0)2 97 87 45 62

A mi padre Julian, a mi madre Lida y mi hermano David, gracias por creer en mi.

Acknowledgements

First of all, I wish to express my sincerest gratitude to my advisors for their guidance, encouragement, patience and, above all, their support during all these years. I want, without a particular order, to express my deepest gratitude to Professor Nubia Velasco because, without her support, this thesis and probably my entire experience in research would not have been possible. You trusted in me blindly in a hard moment and helped me to succeed with my research. Since the beginning, you always pushed me to achieve this goal and found the means to make it possible. In a similar vein, I cannot stress enough how much of a great pleasure has been for me to work with the Dr. André Rossi. He demonstrated exceptional dedication to guide this research, while helping me to conceive new ideas; moreover, his patience to correct my writing and keep me in the right path has been as important for the development of this thesis as my own work. I also owe great gratitude to Professor Marc Sevaux, who has provided me with immense mentorship, not only in guiding my research but by helping me to become a scientist. Last but not least, I wish to thank Professor Eric Bourreau for his contributions to this project. Our collaboration was a rich experience that taught me to work in a team to achieve the goals that might push the limits of science.

Furthermore, I wish to express my recognition to the various members of the Lab-STICC in Lorient and PyLO in Bogotá, whom I have crossed path during these years, for providing me with an excellent atmosphere not only for doing research, but also in a personal basis.

To my parents, too many things to say. Of course I appreciate your endless love and support, but even more I appreciate your faith in me. During my whole life, you motivated me to reach this stage; you always believed that I am better person than actually I am. That explains why I have worked that much to reach that standard and make you feel proud of me, still a lot of work has to be done, but I will keep going on thanks to you.

Of course I also want to thank my friends. You always accompanied me through this process in so many different aspects that I can fill several pages only recognizing your individual contributions. Thanks for the beers we shared, the dialogues, the discussions, the travels, and our long political speeches to expose our arguments and plans to boost our beautiful country to the greatest leagues, someday we will. Thanks for helping me to control my stress, and being always there cheering me up and staying at my side through the good times and the bad ones.

Contents

Table of contents	1
1 Introduction	4
1.1 Motivation	4
1.2 Contributions of this thesis	6
1.3 Outline	7
1.4 How to read this document	8
2 Energy efficient coverage in wireless sensor networks (WSN)	9
2.1 Introduction	9
2.2 Overview of wireless sensors networks	10
2.2.1 Types of networks	10
2.2.2 Sensing models	11
2.2.3 Connectivity	14
2.2.4 Energy consumption in wireless sensors	14
2.3 Lifetime and coverage optimization on WSN	16
2.3.1 Related works	17
2.3.2 Duty scheduling on densely deployed WSN	18
2.3.3 Battery life and network lifetime	20
2.4 Column generation based approaches for wireless sensor networks optimization	21
2.4.1 General model and basic ideas	22
2.4.2 Related problems and directions of research	24
2.4.2.1 Maximum network lifetime	25
2.4.2.2 Partial coverage and lifetime	25
2.4.2.3 Strategies to solve the pricing subproblem	26
3 A numerical evaluation of acceleration strategies for column generation applied to wireless sensor networks optimization	29
3.1 Introduction	29
3.2 Problem description and related work	31
3.2.1 Mathematical approach	31
3.2.1.1 Master problem	32
3.2.1.2 Pricing subproblem	32
3.2.1.3 Convergence of the Column Generation algorithm	33
3.3 Acceleration strategies for the column generation algorithm	35
3.3.1 Dual variables stabilization	36
3.3.1.1 BoxStep stabilization method	36
3.3.1.2 Generalized BoxStep method	38
3.3.1.3 Neame’s stabilization method	40
3.3.1.4 Dual variable values initialization	40
3.3.1.5 Computational experiments	40

3.3.2	Intensification strategies	44
3.3.2.1	κ -Intensification	44
3.3.2.2	Diversification	46
3.3.2.3	Intensification and diversification through a Genetic Algorithm	46
3.3.2.4	Computational experiments	48
3.3.3	Hybridizing stabilization and intensification strategies in column generation	50
3.3.3.1	Computational experiments	50
3.4	Conclusions and future work	51
4	A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints	53
4.1	Introduction	53
4.2	The maximum network lifetime problem under coverage and connectivity constraints	56
4.2.1	Decomposition approach	57
4.2.1.1	Pricing subproblem	57
4.3	Solving the pricing subproblem	58
4.3.1	A GRASP approach to solve the pricing subproblem	60
4.3.1.1	Greedy randomized construction	61
4.3.1.2	Solution improvement	61
4.3.2	A VNS approach to solve the pricing subproblem	62
4.3.2.1	Initial solution	63
4.3.2.2	Local search	63
4.3.2.3	Neighborhoods	64
4.3.2.4	Shake	65
4.4	Computational experiments	66
4.5	Conclusions and future work	73
5	Exact approaches for lifetime maximization in connectivity constrained wireless multi-role sensor networks	74
5.1	Introduction	74
5.2	Problem description	76
5.3	Solution approach	79
5.3.1	Pricing subproblem	80
5.4	Solution approaches to address pricing subproblem	81
5.4.1	A decomposition approach to address pricing subproblem	82
5.4.2	A constraint programming approach to address the pricing subproblem	86
5.5	Computational experiments	88
5.6	Conclusions	92

6	Partial coverage to extend the lifetime in wireless multi-role sensor networks	93
6.1	Introduction	93
6.2	Problem description and model	95
6.3	Solution approach	98
6.3.1	A constraint programming model for the pricing subproblem . . .	99
6.3.1.1	Constraint programming definitions	99
6.3.1.2	Constraint model	100
6.3.1.3	Search strategy	102
6.3.2	An evolutionary algorithm to boost up CG	102
6.3.3	Hybrid CG+EA+CP approach to maximize network lifetime . .	106
6.4	Computational experiments	107
6.5	Conclusions and future work	111
7	General conclusions and future works	115
7.1	General remarks	115
7.1.1	Thesis summary	115
7.1.2	Methodology and findings summary	116
7.2	Limitations of this study	117
7.3	Perspectives of research	119
	References	132
	List of Figures	133
	List of Tables	134



Introduction

1.1 Motivation

Already in 2003, wireless sensor networks (WSN) were promoted as one of the ten breakthrough technologies with potential to change the world [MIT Technology review, 2003]. The research in this field is still growing at a rapid pace and the future continues to be promising. Indeed, recent reports indicate that the world market for WSN is expected to raise from US\$ 2.7 billion in 2012 to U\$12 billion in 2020 [The wall street journal: market watch, 2014]. Moreover, as with most current technologies, probably we have not devised yet the whole range of applications where WSN can successfully provide support. Nonetheless, research on this topic has benefited from large financial support while world leading companies have massively invested in sensor technology for different applications [Tech investor news , 2014].

WSN technology is becoming more and more important thanks to the rise up of new connectivity technologies, and the new practical uses for smartphones and similar devices connected to the Internet of Things. WSN have been successfully deployed in environments where the use of traditional wired networks was too difficult or unpractical [Biagioni and Bridges, 2002, Bokareva et al., 2006, Chen et al., 2013, Kim et al., 2007, Werner-Allen et al., 2006]. WSN present several advantages over wired networks in specific applications. For example, it is worth and useful to detect forest fires *in situ*, as close to their origin as possible in spatial and temporal contexts [Ramsden, 2009]. In general, WSN might be easily adopted to monitor distant and/or hostile environments with sensors that can be remotely deployed and controlled, *e.g.*, by using an airplane, a helicopter or a drone.

Wireless sensors are tiny electronic devices provided with low computational, memory, and communication capabilities that are typically powered from an external, usually limited, energy source. Although processors, memory and radio technology seem to have achieved a mature development stage, the energy sources of wireless sensors are still their biggest weakness. In general, the exploitation of micro-sensors is possible. Nevertheless due to the high cost, or impossibility, of either renewing the energy supply of the nodes, or redeploying the sensor nodes, the constraints faced when designing a WSN system are still severely limiting their usability [Zhang, 2012].

The new advances in WSN technology at different levels have brought many new problems to be studied for researchers in different fields. Routing and communication

protocols, for example, are key aspects in WSN design [Akkaya and Younis, 2005, Li and Bartos, 2013, Younis et al., 2013]. Nonetheless, efficient use of sensors energy is probably one of the major issues to be addressed, specially when it concerns remotely deployed and unattended WSN [Raghunathan et al., 2002]. A large number of different approaches have been proposed to optimize the use of energy stored in the batteries while satisfying some operational constraints, *e.g.*, coverage of some targets regions or points, and connectivity to the base station [Anastasi et al., 2009, Zorbas and Douligeris, 2010].

A vast part of the literature devoted to network lifetime, or the use of energy throughout lifetime, of WSN is based on the use of both heuristic criteria and heuristic algorithms to tackle the problems [Cardei et al., 2005, Slijepcevic and Potkonjak, 2002, Yang et al., 2006, Zhao and Gurusamy, 2008]. Less efforts have been focused on producing global solution approaches able to return optimal solutions for such problems. Moreover, neither global solutions nor methods that guarantee that the optimal solutions are found are available for many problems. As a consequence, even if such methods are efficient and competitive, an accurate evaluation might not be possible and comparisons are often performed against previous proposals rather than to the optimal solution.

The role of global approaches to solve coverage and scheduling problems in WSN has only been slightly explored [Gentili and Raiconi, 2013, Gu et al., 2009b, Liu et al., 2011, Raiconi and Gentili, 2011, Rossi et al., 2012b]. In general it consists on the use of centralized approaches used to define the operation and the structure of the network during its lifetime. Several approximation algorithms have been proposed for metrics as the lifetime, defined as the time until which sensors are able to satisfy the coverage requirements and transmit the information to the base station. Unfortunately, such approximations may lead to suboptimal solutions that are still far from optimal and inefficient from the view point of the use of the energy. Efficient exact approaches have been proposed as well for some specific classes of problems in WSN [Liu et al., 2011, 2009]; nonetheless, some considerations are not general and might require to be adjusted to consider a wider set of characteristics that are possible in the WSN operations.

Column Generation (CG) is one of the most efficient global strategies to compute energy efficient operation schedules in WSN. Based on a single model built over an exponential number of variables, CG offers an elegant and natural method to tackle lifetime and coverage problems in wireless sensor networks. By considering CG, it is possible to represent different characteristics and requirements on the network operation within a single framework that, additionally, can be easily adapted to deal with different objectives. The general idea is to divide the problem into two: *(i)* a restricted master problem (RMP) used to consider the scheduling decisions, *i.e.*, the operational schedule for every single node (or subsets of nodes) during its lifetime and *(ii)* a pricing subproblem (PS) that identifies configurations of the network sensors that might help to extend lifetime or to improve the quality of coverage.

CG iterates between both, the RMP and PS subproblems exchanging the information required to compute the solution that optimizes the desired objective. In the context of WSN the use of CG to maximize lifetime has been recently explored [Gu

et al., 2009a, 2011, Raiconi and Gentili, 2011, Rossi et al., 2012a, 2013, Singh et al., 2013]. It has demonstrated to be an interesting approach that aims to globally solve the problem considering the necessary constraints imposed by the sensor devices and/or by the application. The simplicity of its use in WSN design relies on the fact that the RMP does not need to consider integer variables and, in this way all the pressure to keep the method efficient falls over the pricing subproblem. However, CG has still several drawbacks that may limit its usability. The RMP may be defined over an exponential size set of variables and, in addition, the algorithm can present slow convergence to an optimal solution. Hence, by using CG it can be required to enumerate a large number of columns with the implicit cost of a higher computational effort [Gu et al., 2009a,b]. Moreover, PS may be a difficult problem, what is a drawback considering that it has to be solved at each iteration. Then efficient strategies to produce interesting columns can be necessary with the purpose of exploiting efficiently the CG approach. As expected, the combination of the two characteristics can be catastrophic for the CG performance.

In this thesis, the use of column generation as a general framework to optimize the use of the energy in wireless sensor networks is explored. The study is devoted to the use of CG to solve several versions of the maximum network lifetime problem in wireless sensor networks (MLP) in which sensors are required to monitor discrete targets. The use of CG is evaluated in the context of nodes scheduling where different objectives and requirements for the deployed networks have to be continuously satisfied. Moreover, the difficulties faced when using CG are analyzed in detail to provide helpful strategies that might be more efficient.

1.2 Contributions of this thesis

A central part of this work is dedicated to the study of CG as a framework to solve coverage and scheduling problems in WSN. Along the manuscript, it is demonstrated that this technique can be efficiently combined with exact and metaheuristic approaches to tackle the design of energy-efficient WSN. With this purpose, the use of hybrid approaches combining linear programming, integer programming techniques and metaheuristic approaches in a single framework is studied. The main contributions of this thesis are summarized below:

- It is shown how column generation can be “easily” adapted to tackle a wide range of optimization problems in the context of energy-efficient design of WSN operations.
- The performance of the CG method applied to WSN is analyzed in its simpler form based entirely on state-of-the-art solvers used to tackle mixed and integer representations of the problem. Whenever limitations of these simpler implementations of CG are observed, the causes of poor performance are analyzed in order to identify strategies to overcome the existing difficulties.
- The proposed CG-based method is used to solve several versions of the MLP in wireless sensor networks. Characteristics such as connectivity to the base station

by using energy-efficient connected structures are evaluated. Moreover, it is shown how these approaches can be efficiently adapted to consider extended versions of the problem for which partial coverage of the targets is allowed.

- Several solution strategies based on heuristics and exact approaches are proposed to solve the hard combinatorial problems corresponding to the PS found for the different versions of WSN problems.
- It is shown that by combining efficiently the use of exact and heuristic approaches within the CG framework, this method can be used to compute upper bounds for the lifetime of WSN and to provide a tool to evaluate the performance of the distributed and heuristic approaches widely adopted for the WSN practitioners.

1.3 Outline

- Chapter 2 introduces the WSN and presents some generalities about the operation of these devices. Moreover, an overview of the related work is proposed. The advantages and disadvantages of these approaches are discussed. Finally, the general characteristics of CG to optimize the use of the energy in WSN is provided. The general modeling strategy adopted in this work is introduced and it is shown how it can be adapted to different versions of the maximum network lifetime and related problems.
- Chapter 3 evaluates the performance of CG in the context of WSN. CG is used to solve the minimum coverage breach problem under bandwidth constraints (MCBB), and its performance in this problem is deeply evaluated to characterize its convergence. The use of several general strategies intended to improve the performance of CG, and that can be extended to different WSN problems, is evaluated. The effect of those methods on the number of iterations required to achieve the optimal solution is explored to highlight the advantages and disadvantages. General remarks applied to CG for WSN optimization are also formulated.
- Chapter 4 addresses the maximum network lifetime problem (MLP) in wireless sensor networks with connectivity constraints (CMLP). In order to solve the problem, an exact approach based on column generation that combines heuristic and exact approaches is proposed. Moreover, it is extended to the case of targets partial coverage, where it is not necessary to cover all of the targets but at least a fraction α of them (α -CMLP). A hybrid sequential strategy to efficiently approach a (near)optimal solution for the problem is proposed. Moreover, the use of the techniques presented in Chapter 3 is adapted in this problem to further accelerate the method.
- Chapter 5 extends the use of CG to solve a extended version of CMLP in which sensors can adopt different energy consumption profiles according to the tasks they perform within the network (CMLP-MR). In this section the use of exact

approaches to solve both, the restricted master problem and the pricing subproblem is explored. The use of Benders decomposition and constraint programming approaches are evaluated and compared as strategies to solve the pricing subproblem. Finally, a simple acceleration procedure is presented to help the method to approach optimal or near optimal solutions efficiently.

- Chapter 6 provides a sequential strategy to solve CMLP-MR when partial coverage of the targets is allowed (α -CMLP-MR). An evolutionary algorithm with connectivity and coverage repair operators is proposed to efficiently solve the problem. Then, a constraint programming strategy is added to keep the method operating as an exact approach.
- The conclusion section provides a compilation of the findings of the specific research on WSN addressed in this work. A summary of the results obtained and the challenges faced when CG is adopted to solve the different versions of the optimization problems is also presented. Additionally, the readers will find interesting directions of research to extend the approaches proposed here to a general lifetime optimization problem in WSN considering sensors with heterogeneous and diverse capabilities. Possible extensions of the studied approaches not only to the field of WSN optimization but in general to network design problems are indicated.

1.4 How to read this document

The structure of this manuscript is defined in such a way that the main chapters are self contained so they can be read independently. The latter is a consequence of the fact that Chapters 3 to 6 correspond to published or submitted material for journal publication. Consequently, when reading this document from the beginning to the end, the reader will find inevitable repetitions.

For those readers interested in the whole content of the manuscript, it may be interesting to make a special focus on Chapters 2 and 3 where the background of the adopted approaches is described. Then, these readers may save some time by skipping the introductory sections and skimming reading the generalities of the column generation adopted to solve the problem, as they rely on the same general structure. In contrast, readers interested in the techniques and theory related with some of the specific problems addressed may directly start by reading those chapters. In the latter case, the author will find all the information and a detailed description of the methodology in the required section.

2

Energy efficient coverage in wireless sensor networks (WSN)

2.1 Introduction

Wireless Sensor Networks (WSN) are one of the new promising technologies introduced with the purpose of making easier the collection of data from natural or built environments. WSN are compound by small devices called sensors (or sensor nodes) provided with sensing and monitoring capabilities that are deployed to monitor or control some interesting phenomena. WSN have found space in a wide range of domains. Recently, WSN have shown to be profitable in industrial settings, environmental monitoring, forest fire detection, and some applications where the use of traditional wired networks was complex or even impractical [Biagioni and Bridges, 2002, Bokareva et al., 2006, Chen et al., 2013, Dargie and Poellabauer, 2010, Gaura et al., 2010a, Kim et al., 2007, Werner-Allen et al., 2006].

WSN are a classical example of a network composed by “cooperating objects” that work in collaboration to satisfy an expected result *e.g.*, the monitoring of a certain interesting phenomenon and the communication of the collected information to the final user. While the capabilities of a single sensor might be quite basic, the interest of their use relies on the capabilities and new possibilities obtained by deploying hundreds or thousands of these devices. Sensors can provide monitoring that has not previously been available [Tech investor news, 2013]. They can operate unattended in conditions of temperature, pressure or humidity, where previously was not possible to access with measurement technologies to keep record of needed variables.

Although interesting applications for WSN have been well known for a long time, the rise of WSN technology in industrial or environmental applications is recent. This delay in WSN adoption is probably a consequence of the lack of technology necessary to produce sensor devices at a low cost and the delay in the development of communication protocols appropriate for this kind of devices. Recent advances in material science, networking and semiconductor technologies have empowered the development of the elements required to implement large-scale WSN. Furthermore, new manufacturing process, micro-machining and technology improvements in power sources allow finally to have access to these technologies at an attractive cost.

This chapter aims at introducing some generalities concerning the operation of WSN.

Some technical details that characterize their operation are introduced with special emphasis in those concerning the energy-efficient design. The sources of energy consumption in wireless sensor networks are presented and the abstractions and assumptions considered for modeling along this research work are described. The related work associated to that described along the manuscript is presented, as well as some generalities of the solution approach adopted to solve different versions of coverage and scheduling problems in WSN.

2.2 Overview of wireless sensors networks

As previously mentioned, WSN are composed by a large number of sensors deployed to monitor an interesting phenomenon. Sensors are typically small electronic devices composed by the combination of sensor circuits used to collect information from the environment that might be application-dependent. A sensor node integrates sensing, processing, and communication sub-systems. The sensing subsystem links the sensor to the world and signals that it has to monitor. The processing subsystem provides the sensor with the capabilities to perform medium to high complexity computations and to make low complexity decisions. The communication module consists typically in a radio unit that allows the sensor to communicate with other nodes around it by using short range radio signals. Finally, all of these parts are coupled with a power subsystem in charge of providing the energy required for the sensor nodes to perform their operations.

Sensors are either *passive* or *active* [Sohraby et al., 2007]. Passive sensors are those used to monitor signals as humidity, temperature, vibrations, etc. by using passive measurement sensing units. Active sensors include elements as radars or sonars that might require as much energy as the communication technologies in order to detect the interesting signal from the environment. As expected, this characteristic complicates the network design, as the WSN might require to answer to different energy draining conditions that may heavily affect its performance. Current technology combines different characteristics into a single device that may be used to accomplish different and/or simultaneous tasks before exhausting all of its energy.

2.2.1 Types of networks

The application for which WSN are developed may certainly affect the challenges and constraints faced when planning their operation. Depending on the environment in which sensors will operate, the characteristics of the sensors, their technical specifications, and the strategies adopted to use efficiently their resources may require to be adapted.

WSN may operate unattended in certain applications, *e.g.*, when it is deployed in remote of hostile environments. These *unstructured* networks may be randomly deployed from a remote location where neither the location of the sensors nor the topology of the network are known in advance, before sensors deployment. As a consequence, efficient algorithms, protocols and strategies need to be designed to answer to these uncertain

conditions in order to use the resources efficiently. By contrast, in *structured* networks the location of the sensors can be pre-planned in such a way that the operational constraints of the network and the waste of sensors resources can be avoided a priori, before sensors actual deployment. In other words, the WSN are designed to operate efficiently rather than to respond to random conditions.

Yick *et al.* [Yick et al., 2008] classify the WSN into five types: terrestrial, underground, underwater, multimedia, and mobile WSN. However, the limits between these categories might be unclear and diffuse considering that current sensor technology may embed a lot of different technologies and capabilities into a single node. An overview of the specifications of these categories regarding the structure of the network and energy consumption is outlined in Table 2.1.

Type of WSN	Cost	Numb. of Sensors	Energy consumption	
			Transmission	Sensing
Terrestrial	Low	High	Low - Medium	Low
Underground	Medium - High	High	High	High
Underwater	Medium - High	Low	High	Low
Multimedia	Medium	Low	High	Medium - High
Mobile	Medium - High	Low - Medium	Medium	Low - High

Table 2.1 – A comparison of WSN main families

2.2.2 Sensing models

One of the most important questions in WSN design is the coverage [Cardei and Wu, 2004]. Sensors are deployed to retrieve periodically information either from the environment or from some targets to keep record of the values of some interesting variables. As expected, the coverage strongly depends of the characteristics and technology embedded in the sensors. Sensor nodes may have different types of sensing devices that are selected based on the requirements of the application [Amac Guvensan and Gokhan Yavuz, 2011]. Furthermore, each application may specify different conditions that impose the terms in which coverage is provided.

The coverage that sensors provide strongly depends on the technologies they have embedded and the application for which they are used. Consider for example a video network used for surveillance, in this case the monitored area for each sensor might correspond only to a region that lies within a certain angle and distance. By contrast, a sound sensor may be able to monitor the phenomena occurring within a spheric 3D region around it defined by the maximum sensing distance.

Maybe the simplest coverage model that can be considered is the binary disc model in 2D regions [Zhu et al., 2012]. According to it, each sensor is capable of sensing only from points located within the circular disk around it and located within a defined sensing range R_s . This is probably the most studied model regarding to the coverage in WSN and is the basis for a lot of theoretical advances in WSN theory. As the name indicates, every point in the binary disc model only has two states, covered or uncovered

according to its distance to the nearest active sensor. Nonetheless, this model can be extended to consider probabilistic coverage in which the probability of coverage may decrease as the distance to the active sensors increases [Zhu et al., 2012]. Figure 2.1 depicts the coverage and communication disk model in WSN. The external and internal discs represent the communication R_c and sensing R_s ranges, respectively, for sensor s_1 . Sensor s_1 is able to establish communication with sensor s_2 and is able to survey the point κ_1 . The sensor s_3 is disconnected as it is located out of the communication range. In the same way, the points out of the disk are not covered, *e.g.*, the point κ_2 . Similar considerations may apply for a binary disc connectivity model.

Sensors may be deployed to monitor a continuous area around it, that is probably a representative sample of the space around it, as in the case of temperature or humidity sensors. In this case it is possible to say that sensors are deployed for *area coverage* (see Figure 2.2a), and the interesting area can be seen as a collection of areas, derived from the region covered by each sensor, to be covered. Point or *target coverage* consists in the monitoring of certain points located at discrete positions in the space that may correspond to positions on a grid or randomly distributed in the interesting space (Figure 2.2b) [Cardei and Du, 2005, Cardei et al., 2005, Jia et al., 2008, Rebai et al., 2014]. In *Barrier coverage* the idea is to guarantee that any intrusion to an interesting area is detected, which indicates that emphasis must be addressed to cover any point in the frontier of that region [Kumar et al., 2005]. Finally, in WSN deployed for tracking applications, sensors are at charge of tracing the position of a given target at every moment. *Target tracking* is a typical and challenging application that in addition to identify the set of sensors able to monitor the intrusion, may require the prediction of the movements in order to optimize the use of the energy.

In addition to the previous categories, the applications may impose additional constraints in order to guarantee that the collected information is reliable and/or the quality of the coverage is appropriated. In *Q-coverage*, for example, it is mandatory to cover every target (area) up to Q times, *i.e.*, with at least Q active sensors [Abrams et al., 2004, Singh et al., 2013]. Different reasons may imply that the coverage is not geometric, for example the covered area might not be a perfect disk around the sensor, but a region with a different shape (see Figure 2.2c). In the case of directional sensor networks, the region covered by the sensors may be a conic region or a semicircle in the 2D space (see Figure 2.2d). All of these considerations may affect the performance of the sensors and the operations planning in WSN; however, in practice, the modeling can be simplified if it is possible to assume that every sensor is aware of the regions (targets) that it covers. A complete review of coverage problems in WSN is presented by Li and Liu [Deying and Liu, 2009].

Relations between area coverage and discrete target coverage are well known [Gallais et al., 2008]. For example, area coverage may be represented by using discrete points representing either the regions covered by the same set of sensors or the intersection between the frontiers limiting the covered area. This kind of relationships can be exploited when area coverage problems are considered in such a way that approaches used for discrete coverage can also be used to approximate the coverage of continuous areas.

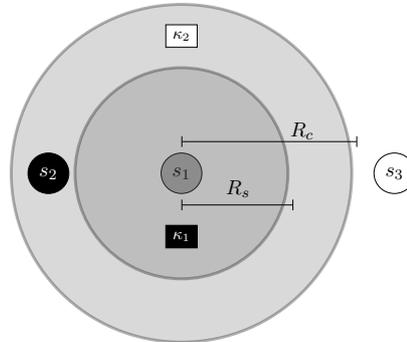
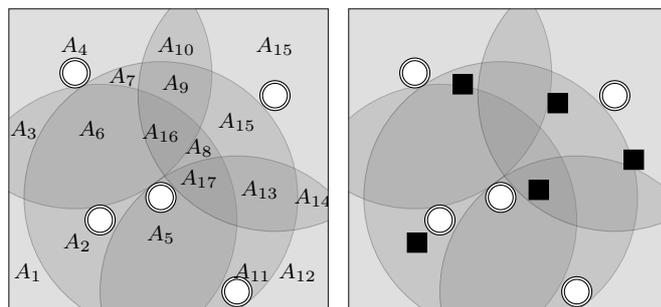
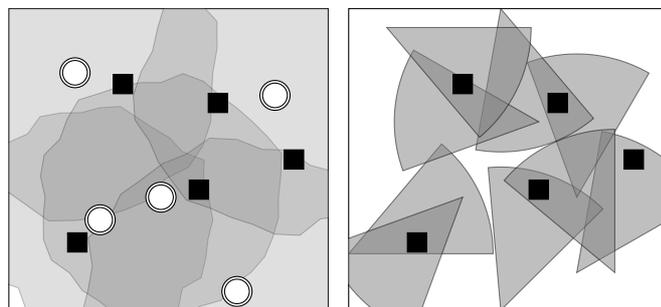


Figure 2.1 – A simple coverage and connectivity model depicting the sensing and communication ranges



(a) Area coverage with sensors (b) Target coverage with sensors



(c) Irregular sensing area (d) Directional sensors

Figure 2.2 – Different models for coverage in WSN

2.2.3 Connectivity

If sensors are not provided with storage capabilities, are located distantly, or the information must be addressed on time when it is available, connectivity represents an important issue in WSN. The connectivity requirement is met when every active sensor is able to find a path to send data to the final user or sink. Sensors use their transceiver units to communicate the information they collect or retransmit the information received by other sensors on the network. In general, communication is addressed to a final user or sink via multi-hop wireless communications. Consequently, in order to establish communication and transfer the information, each sensor on the network requires to be able to find a path to the sink for the information it collects. If such a path is not available, the sensor is isolated and is useless within the network.

In general, connectivity between sensors is achieved through wireless signals that are not necessarily directed but scattered in the space through an undirected antenna. Thus, connectivity is obtained through a virtual backbone [Lu et al., 2005] representing these virtual communication links, and the paths used to address the information to the final user (Figure 2.3). Those paths may require to be multiple (m -connectivity) in order to improve the resilience of the network and make it fault tolerant. If this is the case, the network structures will be more complex than a tree backbone.

Relations between coverage and connectivity have been previously described. Tian *et al.* [Tian and Georganas, 2005] demonstrate that in sensors deployed to accomplish area coverage, connectivity is guaranteed if the transmission range is at least as big as twice the sensing range. However, the same observation is not valid anymore when discrete target coverage is considered [Lu et al., 2005]. In addition, this assumption is not always realistic and, as a consequence, current approaches to guarantee area coverage and communication need to consider both requirements simultaneously.

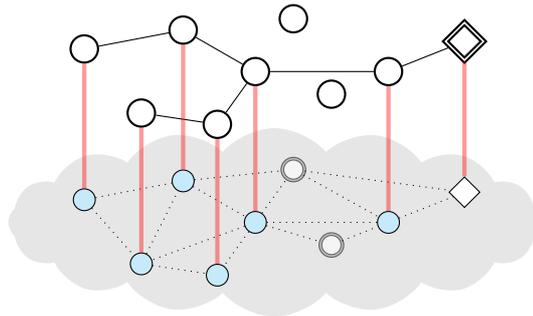


Figure 2.3 – Virtual backbone

2.2.4 Energy consumption in wireless sensors

Energy consumption in WSN strongly depends on the characteristics of the sensor node [Anastasi et al., 2009]. Indeed, recent research demonstrates that there exist huge differences in the energy consumed by different commercial nodes [Raghunathan et al., 2002]. It has been claimed that certain remarks remain present almost in every current

sensor node, however, no agreement on this subject has been reached yet. In general, it is assumed that the communication subsystems account for the largest portion of the energy consumed by sensor nodes [Anastasi et al., 2009, Farahani, 2008]. However, in certain applications the sensing, the signal processing and the hardware operation consume an important amount of power as well [Puccinelli and Haenggi, 2005].

The function of the sensor nodes in the WSN consists in detecting the interesting phenomena, processing the collected information, and (re)transmitting the collected data. Thus, consumption of energy at the sensors is consistently related with these activities [Sohraby et al., 2007]. Power consumption in WSN mainly comes from three factors: communication, sensing and computing.

- **Communication:** It consists on the energy consumed by the transmission and reception modules embedded in the sensor. In general, sensors employ low energy consumption devices; nonetheless, it generally accounts for the bigger portion of the energy consumed. Moreover, the intensity of the use can be an important source of consumption and networks with higher sampling rates necessarily drain the energy faster than networks that sample occasionally.
- **Sensing:** Different applications may imply a lot of differences in the energy consumption rates associated with the sensing activities. Sensors may be used to monitor easy variables as the temperature, which only require the use of a passive device with a low energy consumption rate. However, some applications (for example the ultrasonic sensors) may require the sensors to generate signals and capture the answer, which can lead to a higher energy consumption.
- **Computing and information processing:** This is the energy consumed by the sensors to perform data processing and decision making tasks. Current technologies allow sensors to perform even complex computation tasks; nonetheless, it means that energy expenses may increase. Each sensor receiving data either raw from the environment or processed originated in other sensors might be required to encode and create packets with the information at expenses of higher energy consumption rates.

In multi-hop WSN, the communication task certainly implies harvesting, processing and re-transmitting the information collected by other sensors, consequently the power consumption could increase depending on the traffic of the network.

The energy consumption rate associated to a sensor node is also related with the activity that it performs within the network. The rate at which power is drained from the sensors might not be constant; consumption rates may depend on the different roles assumed by the sensors, and that can be used to save energy. Some operating modes (roles) that they can adopt can be classified into the four following categories [Zhu et al., 2012]:

- **On-duty:** All the components of the sensors are operative in order to collect information about the interesting variables, process the information, perform any

type of computation and (re)transmit the information to other sensors or the final user.

- **Used for transmission:** In this case the sensor is only used to re-transmit the information collected by other sensors. It is not retrieving any information from the environment but still has to perform any type of computation to transmit the information it receives. The sensor is only used to help keeping connectivity within the different parts of the network and providing a path to send the information to the final user.
- **Used for coverage:** Sensors can turn off the communications technologies in order to save energy and avoid redundant information interfering in the networks. Sensors can activate their communication modules as a response to a phenomenon detected or may store the information in a memory unit, if sensors are provided with it, to transmit it just in the right moment when it is required.
- **Off-Duty:** The sensor is in an idle state in which it is neither used for communication nor for sensing purposes and it consumes energy at a negligible rate (typically some self-discharge is observed). The sensor may have some mechanism that will reactivate the sensor once it is required.

As expected, each of the modes above consumes power at different rates depending on the sensor modules involved in the operations. Moreover, additional variables may influence the power consumption of the sensors. The power consumed might increase as a consequence of the traffic that passes through a sensor node due to the amount of transmissions established and the processing, packing and retransmission of the information. Similarly, the energy consumed by transmission may depend on the distance to the receptor node (sensor or base station). A classical example of sensors draining energy at different rates is observed in sensors that can adjust their sensing or communication ranges.

2.3 Lifetime and coverage optimization on WSN

Several studies have been devoted to represent all of the aspects of the energy consumption, the network operations, and the lifetime in WSN. Different characteristics and specifications of the network operation have been provided, some of them designed with the specific purpose of network optimization in one of the large number of aspects involved in WSN design. WSN are resource constrained devices that impose big challenges to overcome the operational constraints and successfully deploy these devices without neither affecting their reliability nor wasting their energy. The use of energy and the knowledge of its consumption is a major concern to be considered when dealing with WSN design.

Lifetime is probably one of the most studied metrics in WSN; however, its definition might be flexible and application specific [Sha and Shi, 2005]. A lot of definitions have been proposed according to different specifications and requirements for the network

operations. It was classically defined as the time until which the first node fails [Shi, 2007]. Nonetheless, this definition does not take into account that WSN can be still fully operative and able to survey the required phenomenon after some sensors fail, *e.g.*, densely deployed WSN for temperature monitoring. Consequently, several definitions based on the availability of nodes, the sensor coverage, and the connectivity have been also explored. For a complete review of definitions of network lifetime in WSN, the reader may be referred to the manuscript of Dietrich *et al.* [Dietrich and Dressler, 2009].

In the context of this work, *network lifetime* is defined in terms of sensor coverage and network connectivity. It is defined as the time interval during which sensor network can perform the sensing functions and is able to transmit the collected information to the sink used to compute or retransmit the information [Cardei and Wu, 2004]. In other words, the lifetime of a sensor network is the total time that the WSN have sensors with the necessary energy to provide the required level of coverage of the interesting phenomena and transmit the information to the sink. This definition does not make any assumption about the characteristics of the coverage that has to be delivered. Consequently, it is possible to extend this definition to different scenarios, *e.g.*, when sensors are required to overcover targets by using several sensors, sensors are heterogeneous, or when sensors are used to survey a 3D region.

2.3.1 Related works

WSN technology is still on a development phase and requires to be optimized for a successful adoption. Some works are devoted to improve the performance of WSN not only in the technical aspects, the technology embedded, their sensing capabilities, and their power sources, but also to take the maximum advantage of the possibilities they offer and the limited resources they have.

Several metrics have been defined to evaluate the lifetime and the efficiency of the energy usage in WSN [Dietrich and Dressler, 2009]. Lifetime and coverage are typically related objectives that are commonly used to define network operation in WSN. Whereas it is interesting to provide reliable and timely coverage, it might be also important to guarantee the operation of the network as long as possible. As could be expected, network lifetime strongly depends on the battery lifetime of each individual node. Coverage depends on the availability of sensors; moreover, reliable coverage might require the use of extra sensors that can reduce network lifetime.

According to the capabilities of the sensors, the purpose of the network, or the inner structure of the device, it is possible to create a wide classification of WSN and its energy usage. Anastasi *et al.* [Anastasi et al., 2009] present a comprehensive review of energy conservation schemes on WSN. The authors consider approaches for energy conservation based on power management on sensor nodes as well as approaches based on data acquisition and energy-efficient protocols. From the viewpoint of coverage and connectivity, the applications of WSN can be classified as: *(i)* Area coverage, *(ii)* Target coverage, and *(iii)* Target tracking. Zorbas *et al.* [Zorbas and Douligieris, 2010] present a survey considering the main characteristics studied by researchers regarding

the energy efficiency of WSN used for target coverage. If a continuous region requires to be monitored, it has been demonstrated that area coverage can be accurately represented by discrete points corresponding to the intersections of the circles delimiting the covered area for each sensor [Gallais et al., 2008].

Most approaches proposed to efficiently use sensors energy in WSN rely either on power aware configuration of the networks or efficient scheduling of sensors operations. In the former case, the objective is typically to minimize the energy consumption associated to communication, coverage and computation by finding energy efficient network topologies involving all the sensors [Abdulla et al., 2012, Berman et al., 2005, Santos et al., 2012]. In this way, longer lifetimes are achieved by avoiding misusing the energy in tasks that are not necessary or use it inefficiently inefficient. This approach is particularly useful when sensors are not densely deployed and are required to be active most time throughout network lifetime. In the second case, sometimes called duty scheduling, the objective is to allocate tasks to the sensors and timing for these operations during network lifetime. This strategy is typically adopted when more sensors than required to satisfy the coverage and connectivity constraints are available, *e.g.*, remote or hostile environments where it might not be possible neither to decide the placement of the sensors *a priori* nor to replace sensors batteries after they are deployed [Bokareva et al., 2006, Castaño et al., 2014, Rossi et al., 2012a].

While power aware configuration and routing schemes try to extend network lifetime by minimizing the energy consumption on nodes, this approach might still have some problems. In power aware schemes the decisions considered can lead to overload some particular sensor nodes which typically lead to network lifetime shortages. This problem can be even worst if network lifetime is bounded by the sensors located one-hop away from the base station; an incorrect use of energy can lead to isolate the base station from the network and make the network useless (see *the Hot-spot problem* [Abdulla et al., 2012]). In contrast, approaches pointing to duty scheduling typically solve the problem through global decisions that ensure that power management decisions on nodes are aware of this characteristic.

In densely deployed WSN, it is possible to satisfy the coverage and communication requirements by activating only a subset of sensors. Then, it is possible to extend the lifetime by activating sequentially these subsets at different moments in time in such a way that operational constraints are continuously respected. This approach, known as duty scheduling is focused on the strategic planning of the network operation. The general idea is to put a subset of sensors in active mode, while ensuring coverage and connectivity to the sink, and to schedule subsets of active sensors over the time [Cardei et al., 2005, Castaño et al., 2013, 2014, Lu et al., 2005, Zorbas et al., 2010].

2.3.2 Duty scheduling on densely deployed WSN

In order to extend the lifetime, a proven useful approach is to divide the network into a set of cover sets (or feasible subgraphs) and to schedule them [Cardei et al., 2005, 2006, Gentili and Raiconi, 2013, Gu et al., 2011, Lu et al., 2005, Singh et al., 2013]. Then, only sensors in the active set operate while the rest are inactive. Two main approaches

have been adopted, *disjoint* and *non-disjoint* sets. In the former case, a selected subset of sensors do not share nodes in common with any other throughout network lifetime, *i.e.*, it is used until the nodes in that structure run out of energy. As expected, only one cover set is active at any time and the remaining ones might be in a sleep state in which the sensors do not consume energy (or consume at a very low rate). In the latter, it is allowed to use sensor nodes in different network structures that are scheduled to optimize the use of energy while considering sensors batteries' constraints.

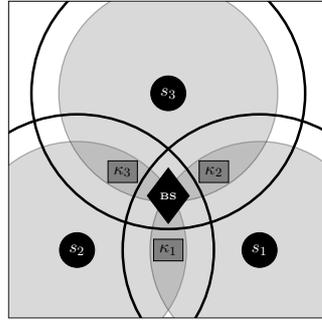
The basic ideas behind duty scheduling were first introduced by Slijepcevic and Potkonjak [Slijepcevic and Potkonjak, 2002], who propose the use of disjoint subset of sensors to extend the lifetime. They propose a heuristic approach to maximize the number of mutually exclusive subsets that will operate until sensors deplete their battery. This concept has been remarkably exploited by researchers in the field of WSN optimization and has conducted to a wide spectrum of similar approaches considering a high number of special characteristics of the network operation [Cerulli et al., 2012, Dhawan et al., 2006, Gu et al., 2009a, Jia et al., 2008, Rossi et al., 2013, Wang et al., 2009].

The use of disjoint cover sets has been largely explored by researchers, and efficient approaches have been proposed to tackle different problems [Ahn and Park, 2011, Cardei et al., 2005, Chen et al., 2010, Slijepcevic and Potkonjak, 2002]. The use of disjoint sets offers robustness on the network operations in the sense that an specific node only belongs to a subset and its failure does not affect different subsets, and only affects the quality of the coverage provided for the current set. Nonetheless, this approach can lead to shorter lifetime as decisions taken are not globally directed to maximize lifetime but the number of cover sets.

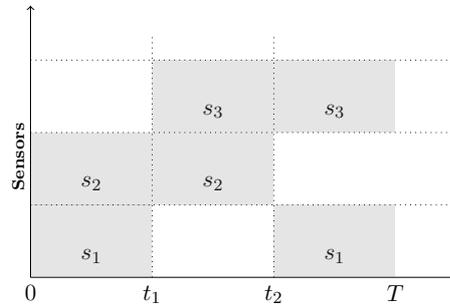
Recently, the adoption of non-disjoint approaches to maximize the lifetime in WSN while considering additional characteristics of the network operation has received an increasing attention [Cardei and Du, 2005, Raiconi and Gentili, 2011, Wang et al., 2009]. In this case the objective changes from maximizing the number of subsets to the identification of optimal schedules for subsets of sensors while respecting the battery capacity and the network constraints. In some cases, the problem can be efficiently solved in polynomial time [Liu et al., 2011, 2007]; however, it largely depends on the characteristics of the network that are considered and in most cases the problems involving the optimal use of energy in WSN require to face hard combinatorial problems. Recently, Gu *et al.* [Gu et al., 2011] demonstrate that, in general, the optimal schedule of sensor nodes in target coverage problems can be represented by using patterns (indicating the energy consumption rate of each sensor) and lately propose a solution method based on column generation (CG) to address a lifetime problem in WSN [Gu et al., 2009c].

In Figure 2.4a it is depicted the idea of non-disjoint subsets previously discussed. Three sensors $\mathcal{S} = \{s_1, s_2, s_3\}$ are used to monitor three targets $\mathcal{K} = \{\kappa_1, \kappa_2, \kappa_3\}$ and send the collected information to the base station BS in the middle of the figure. The inner circle represents the area within which a target can be considered covered by a sensor, and the external circle represents the communication range to establish communication between sensors and the BS. As observed, none of the sensors is able to monitor all the targets at the same time. Consequently, if the three of them have to

be covered, it will be necessary to turn on several sensors simultaneously. By contrast, activate all of them at the same time is not an intelligent strategy, as it consumes more energy than actually required to provide coverage of the targets. If each sensor battery capacity is assumed to provide energy to run the sensor 1 unit of time while used for coverage and communication purposes simultaneously, it is possible to extend the lifetime by activating subsets of sensors in rounds. So, sensors s_1 and s_2 can be active for $t_1 = 0.5$ time units, then sensors s_2 and s_3 can operate for additional $t_2 = 0.5$ time units and the battery of s_2 has been depleted; nonetheless, sensors s_1 and s_3 have still 0.5 time units of remaining energy that can be used to extend the coverage time up to 50% compared to the initial proposal (see Figure 2.4b).



(a) A simple WSN



(b) Optimal schedule

Figure 2.4 – Optimal schedule for the operation of a simple network

2.3.3 Battery life and network lifetime

Gu *et al.* [Gu et al., 2011] present an interesting and simple model, assuming the binary disc coverage model that represents the lifetime of a WSN and is independent of the inner characteristics of the network operations. The latter model is adopted as the basis for the use of column generation and, indirectly, guides the research presented in this work. Consider a set $\mathcal{S} = \{s_1, s_2, s_3 \dots s_m\}$ of sensors deployed to monitor a set of targets (points or regions) $\mathcal{K} = \{\kappa_1, \kappa_2, \kappa_3 \dots \kappa_n\}$. Each sensor is provided with an initial battery charge b_{s_i} and is aware of the targets it covers. Let $U_{\kappa_i} \subseteq \mathcal{S}$ denote the set of

sensors that can monitor the target κ_i . The indicator function $x_{s_i}(t)$ is set to 1 if sensor s_i is active at time t . The function $e_{s_i}(t)$ indicates the total energy consumption rate assumed for the sensor s_i to perform the coverage and communication tasks. If coverage of the targets needs to be guaranteed at every moment during network lifetime, it is possible to represent it through the set of constraints:

$$\sum_{s_i \in U_{\kappa_k}} x_{s_i}(t) \geq 1 \quad \forall \kappa_k \in \mathcal{K} \quad (2.1)$$

Additionally, it is possible to consider the battery limitations for each sensor until network lifetime (T) is reached by representing the cumulative energy consumption through the following constraint:

$$\int_0^T x_{s_i}(t) e_{s_i}(t) dt \leq b_i \quad (2.2)$$

Any schedule for the sensors activation must guarantee that both constraints above are respected, no matter what the objective is. Additionally, extra features such as the connectivity, partial coverage of targets to extend lifetime, or extra requirements of coverage can be introduced by either modifying the previous equations or adjusting additional ones.

Although complex models can be used to represent the sensors' energy reservoirs during its lifetime, simplifications considering *battery efficiency factors* are typically adopted to estimate the actual lifetime of the battery [Farahani, 2008]. Battery efficiency factors can be used to estimate the actual duration of the battery and, finally, to construct accurate lifetime models. Consider Figure 2.5 which represents the instantaneous energy consumption rate and the remaining energy of sensor s_1 in Figure 2.4. Sensors are able to establish reliable coverage and communication only when the remaining energy is above certain threshold level (TH). If the available energy is below that threshold, the sensors might fail and their appropriate operation cannot be guaranteed anymore. The Figure 2.5 presents a simple linear consumption model, the energy is consumed at a constant rate when sensors are active and, consequently, the energy stored decreases linearly when the sensor is providing coverage and/or communication capabilities in the network.

2.4 Column generation based approaches for wireless sensor networks optimization

This section introduces some generalities of the maximum network lifetime problem in WSN. The model will be defined for a generic problem without making any assumption on the energy consumption models adopted for that problem. Then, the characteristics of the method proposed to solve the problem will be presented and it will be indicated how it can be extended to solve other related problems addressed in this work.

As it was mentioned in the introductory section, Column Generation (CG) has been proposed as an efficient strategy to solve coverage and scheduling problems in WSN.

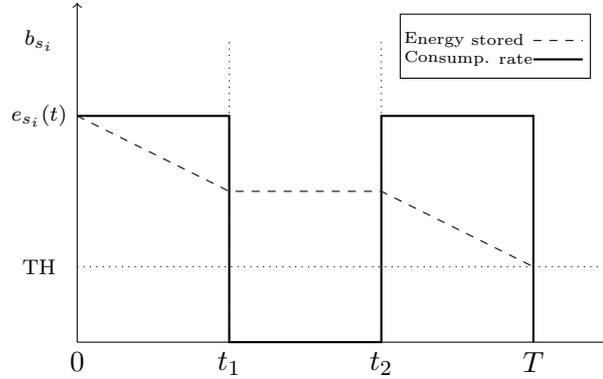


Figure 2.5 – Energy consumption and remaining energy during sensor lifetime

This strategy decomposes the problem in two: (i) a restricted master problem (RMP), and (ii) a pricing subproblem. RMP is used to solve the original problem over a reduced part of the solution space expressed by a subset of columns (variables). The PS is used to identify additional columns that can bring new information to RMP and can be exploited to further improve the objective function of RMP. CG operates iteratively; at the first iteration RMP is constructed over the reduced set of columns and new interesting columns are added by using any possible method useful to solve PS. At each iteration RMP is solved, and the dual variable values associated to the optimal solution are used to check, through the reduced cost criterion, whether or not it exists a new profitable column to improve the objective function. If a new column is available, it is added to RMP and a new iteration is performed; otherwise, the algorithm finishes. If such a column does not exist, it is possible to guarantee that current solution is optimal. A general overview of the CG algorithm is depicted in Figure 2.6.

2.4.1 General model and basic ideas

Consider a set $\mathcal{S} = \{s_1, s_2, s_3 \dots s_n\}$ of sensors used to monitor some interesting phenomena. Let $G(\mathcal{S}, \mathcal{A})$ denote a graph where \mathcal{A} is the set of arcs used to indicate whether or not communication is possible between the two sensor nodes $u, v \in \mathcal{S}$. A communication link between u and v can be established when the node v is located within the communication range R_c of node u . In order to have symmetric connectivity, it is required for node v to be located within the communication range of u ; otherwise we say that asymmetric connectivity exists. An arc $a(u, v) \in \mathcal{A}$ exists only if the nodes u and v can establish a communication link. A subgraph $G'[\mathcal{S}', \mathcal{A}']$, where $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{A}' \subseteq \mathcal{A}$, is feasible if it can satisfy the purpose for which the WSN is deployed, *i.e.*, coverage, while satisfying some specific constraints as connectivity to the base station, multiple coverage of the interesting phenomena, network resiliency among others. The subgraph $C_j \subseteq G$ is the feasible subgraph obtained after allocating energy consumption rates to the nodes to perform the coverage and connectivity requirements. The set of all feasible subgraphs is denoted by Ω .

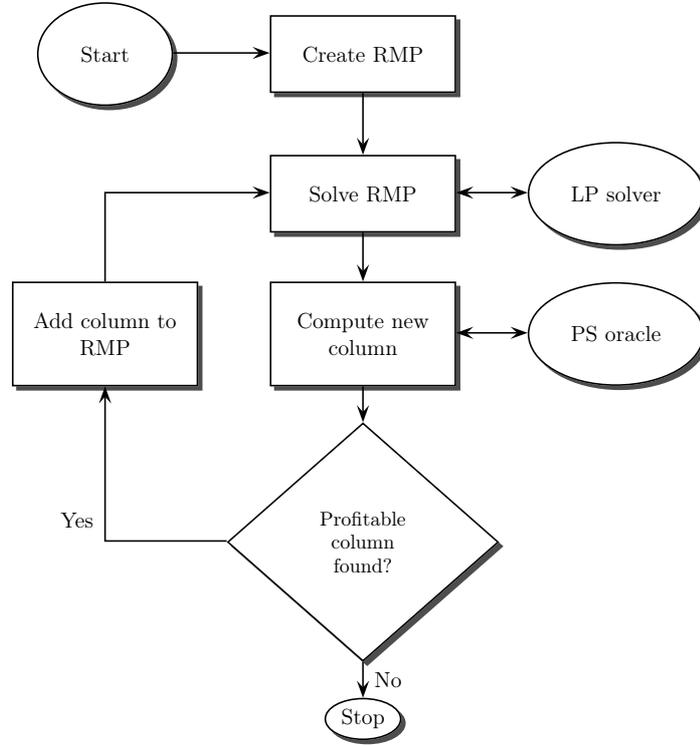


Figure 2.6 – Overview of the column generation algorithm

The total energy consumed for a sensor node in the subgraph C_j can be computed as the sum of the individual energy consumption associated to communication tasks E_{vj}^t , sensing tasks E_{vj}^s , the energy required to keep the sensors operating E_{vj}^a and the self-discharge rate of the sensor when it is idle E_{vj}^i . So, energy consumed depends on the task that sensors performs within the network. Let t_j denote the time interval allocated to the subset $C_j \in \Omega$, the Maximum Network Lifetime Problem in WSN can be defined as finding the optimal pairs (C_j, t_j) such that the battery lifetime of sensor nodes is respected. Assuming that the set Ω is known, the optimal lifetime, and the time intervals that each subgraph $C_j \in \Omega$ is used can be computed through the following model:

The maximum network lifetime problem (MLP)

$$\text{Maximize: } \sum_{j|C_j \in \Omega} t_j \quad (2.3)$$

$$\sum_{j|C_j \in \Omega} (E_{vj}^a + E_{vj}^s + E_{vj}^t + E_{vj}^i) t_j \leq b_{s_v} \quad \forall s_v \in \mathcal{S} \quad (2.4)$$

$$t_j \geq 0 \quad \forall j|C_j \in \Omega \quad (2.5)$$

The objective is to maximize network lifetime (Eq. 2.3), *i.e.*, the time until which network can guarantee the operational constraints by using only feasible network configurations (coverage, connectivity, etc.). Constraints (Eq. 2.4) are used to guarantee that sensors are only scheduled to operate respecting their battery limits. Of course, time variables are linear and continuous (Eq. 2.5).

2.4.2 Related problems and directions of research

Although the model above might be easily solved by using state-of-the-art optimization solvers, the overall problem might require a bigger effort. As expected, by assuming that the set Ω is available, it is assumed that an exponential set of variables is available. However, most times it is not the case and the full enumeration of the elements of such a set is not practical. Moreover, even if somehow they can be enumerated, only part of them will be part of the optimal solution [Dantzig, 1963]. As a consequence, the use of the CG framework previously discussed can be a valuable strategy to address the problem efficiently by enumerating only those columns that look promising to improve the objective function.

Recent research has demonstrated the efficiency of CG to address different lifetime problems in WSN. This approach has been shown to be successful even when new considerations and models regarding the energy usage of these devices are included. It is precisely in that flexibility that relies the interest of the use of CG to solve coverage and scheduling problems in WSN. CG can be “easily” modified to represent different requirements on the network operations, the new inner characteristics of the energy consumption, or the specifications about network structure.

This work makes emphasis on the use of CG to solve two related problems in WSN design. The first concerns the use of CG to solve optimization problems in WSN deployed to provide coverage of discrete targets. This research is mainly focused on the case where information is required to be transmitted to the base station and the cover sets correspond to a connected structure. The second case corresponds to WSN in which partial coverage of targets is allowed as a strategy to extend the lifetime. The latter case includes as well WSN in which lifetime is more important than any other metric and the purpose of the network is to monitor the environment as well as possible [Thai et al., 2008, Wang et al., 2009].

2.4.2.1 Maximum network lifetime

CG has been recently explored in the context of WSN optimization to solve coverage problems in which sensors have different characteristics. This fundamental result is supported by Gu *et al.* [Gu et al., 2009b] who evaluate the use of CG to solve several versions of the maximum network lifetime problems in WSN. Nonetheless, the author's conclusions seem to indicate that even if CG is an interesting approach to tackle this problem it evidences some limitations when complex characteristics of the network are included.

The convergence of CG is the main problem observed, and the authors indicate that it might be necessary to use a heuristic criterion to decide whether or not to stop the optimization process. Raiconi and Gentili [Raiconi and Gentili, 2011] evaluate the use of CG in a WSN problem where connectivity to a base station is required. The authors results confirm that the basic implementation of CG might be inefficient and can be easily outperformed by using global heuristic approaches. If connectivity is not considered, the use of a pure CG problem using ILP to address PS can be efficient enough in certain problems [Cerulli et al., 2012, Gu et al., 2009c, Rossi et al., 2012a, Singh et al., 2013]; however, the complexity of PS increases as additional characteristics are considered and the use of sophisticated approaches seems to be necessary.

2.4.2.2 Partial coverage and lifetime

In some applications, it is possible to cover only partially the interesting phenomena and still be able to accurately estimate the values of the required variables [Gentili and Raiconi, 2013, Li et al., 2011, Liu and Liang, 2005], *e.g.*, in environmental temperature monitoring. If this is the case, it is possible to neglect certain targets or regions as a strategy to extend network lifetime. In other applications it might be more important to guarantee a minimum network lifetime than it is to cover fully the interesting phenomena (or the interesting targets). These two problems might be easily captured and represented under the CG framework studied in this work.

Previous attempts to solve this problem are presented by Wang *et al.* [Wang et al., 2009], Chen *et al.* [Cheng et al., 2005] and Rossi *et al.* [Rossi et al., 2012a] who addressed particularly two target coverage versions of the problem namely the minimum coverage breach problem under bandwidth constraints (MCBB) and the maximum network lifetime problem under bandwidth constraints (MNLB). In the former, the purpose is to guarantee that coverage is provided during at least a minimum lifetime T_0 by using WSN that are constrained in the bandwidth, *i.e.*, the number of sensors that can collect and transmit information simultaneously. This constraint in the number of active sensors may imply that all targets cannot be covered simultaneously, so a global measure that computes an average of the total uncovered targets during lifetime, called the *breach rate*, is defined and is used as the objective to be optimized. In MNLB, a limit on the value of the breach is established and the objective is to maximize the lifetime keeping this value under the desired level of coverage.

Unlike the MCBB and MNLB, where a minimum number of targets covered is not obligated, it is possible to consider an additional version of the problem that obligates

to cover at least a fraction $\alpha \in [0, 1]$ of the targets at every moment during the whole network lifetime. In this way, it is possible to take advantage of the reduced levels of coverage to extend the lifetime considering that exists a clear trade-off between those objectives. Figure 2.7 presents a BoxPlot illustrating the relations between the value α and the increment in lifetime (%). As observed, the variations on lifetime are interesting even for high values of α demonstrating the importance of considering these problems in WSN design when it is possible.

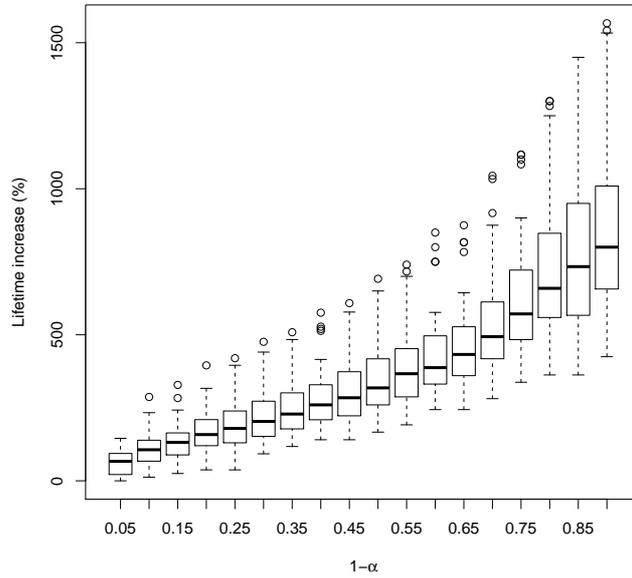


Figure 2.7 – Relation between coverage and lifetime in WSN

In order to adopt the CG framework to solve these two problems, small modifications of the method must be performed. In the case of MCBP, lifetime can be included as an additional constraint in the model (2.3-2.5) and the objective function must be addressed to minimize the breach rather than maximize the lifetime. In the case of MNLB, an additional constraint considering the uncovered targets can be added whereas the objective function (Eq. 2.3) remains untouched. Finally, in the problems in which partial coverage is allowed, the model does not need to be modified; however, in general, the structure of the columns must guarantee that the operational constraints of the network, the energy consumption model and additional requirements are considered.

2.4.2.3 Strategies to solve the pricing subproblem

Even the simplest version of the maximum network lifetime problem may imply that a new hard combinatorial problem needs to be solved at each iteration of CG. In the case, where connectivity is neglected and sensors assume a unique role, the PS corresponds

to a weighted set covering problem, one of the 21 Karp's problems known to be NP-Hard [Karp, 1972]. Even if efficient strategies have been already developed to address some combinatorial problems, this is not always the case. Indeed, by considering the connectivity requirement the effort required to solve PS is largely increased [Klein and Ravi, 1995]. As a consequence, the use of more sophisticated strategies to address PS and accelerate the convergence of CG must be developed to keep this approach operating as an efficient strategy.

In this work, we adopt three different strategies to improve the performance of CG when applied to WSN optimization. Along the manuscript it is shown how it is possible to take advantage of the characteristics of the models used, the efficiency of the heuristic procedures and the information provided for the PS to boost up the CG method.

Heuristic approaches Taking advantage of the fact that CG does not need a proven optimal solution to improve the objective function of RMP, the use of efficient heuristic methods is a natural choice when designing CG algorithms. The use of hybrid strategies combining CG with other approaches has been largely explored [Blum et al., 2008, Talbi, 2013]. For example, in the context of vehicle routing problems CG has been successfully combined with heuristic approaches where columns represent abstractions of the routes and the oracle is used as a route generator taking advantage of interesting advances in methods to solve routing problems. Nonetheless, the drawback of this approach relies on the fact that, without using an exact approach to solve PS, the global approach is a heuristic and cannot guarantee that a solution of RMP is optimal. As a consequence, even if a good method is used to compute solutions for PS, at least one iteration must be performed with an exact approach to keep CG operating as an exact approach.

In this work, the reader will find that several heuristic approaches are either necessary or worth as strategies to solve PS efficiently and improve the performance of CG. Several problems addressed in this manuscript are successfully tackled with heuristic approaches. Problems as the weighted set covering [Chvátal, 1979], the node weighted Steiner tree [Agrawal et al., 1991, Guha and Khuller, 1999] and other related network design problems are addressed in this work and the heuristic approaches are further exploited to help improve CG performance.

Exact approaches The use of efficient exact approaches to solve the PS is necessary to guarantee that solutions found through CG are optimal. In some cases it is possible to take advantage of the structure of PS to solve it efficiently through exact approaches that, additionally, help to keep the global solution method as an exact approach. If exact methods and heuristics are available, it can be worth to take advantage of the best characteristics they offer by combining them in a single method. This approach is specially useful when the performance of the heuristic is weak and it is difficult to improve, yet it produces suboptimal solutions fastly for the PS. Then, each time it fails to find a profitable column, it is replaced by an exact approach which might be required several times before reaching the optimal solution. As a consequence, the performance of such an exact method used to address PS is key for the success of the adopted CG.

This research explores the use of exact approaches to address PS as a unique strategy. Thus, CG is combined with tools as Benders decomposition and branch-and-cut to solve the implicit network design problems obtained as subproblems in the proposed CG method. Moreover, interesting approaches combining CG and constraint programming are implemented and proved to be very efficient.

Accelerating Column generation It is well known that column generation might present some problems that may limit its usability. Additionally to the difficulty of the PS observed in certain cases and addressed through the use of the heuristic and exact approaches previously described, CG may present some problems typical of the structure of the approach. Convergence is an important issue regarding the implementation of CG. While in the first iterations the evolution of the objective function is fast, in the latter iterations these improvements may be marginal and require a large number of iterations to compute the optimal solution (*tail-off effect*) [Gilmore and Gomory, 1961, 1963]. Several causes have been identified for such problem, however, the research points to indicate that it is mostly a consequence of the instability of the dual variables [Gilmore and Gomory, 1961, Lübbecke, 2010]. Additionally, it can appear a *heading-in effect* when implementing CG that can largely affect its performance [Vanderbeck, 2005]. It consists in the successive enumeration of irrelevant columns at the first iterations of CG while not enough information is available to produce interesting columns.

In order to successfully use CG, several methods have been proposed to overcome the typical problems affecting the convergence of CG [Ben Amor et al., 2006, Du Merle et al., 1999, Elhallaoui et al., 2008, Marsten et al., 1975, Rousseau et al., 2007, Wentges, 1997]. The use of some of these techniques is explored in this work in the context of CG. Moreover, it is explored the use of methods returning several and diverse columns at each iteration of RMP to help accelerate the convergence of CG [Moungla et al., 2010]. Finally, a simple technique used to avoid the heading-in effect is adopted to improve the performance of the CG method.

3

A numerical evaluation of acceleration strategies for column generation applied to wireless sensor networks optimization

This chapter addresses a specific problem presented in Wireless Sensor Networks, the minimum coverage breach problem under bandwidth constraints. To solve it, a column generation approach is adopted. Column generation is known to present issues that may lead to slow convergence. This phenomenon used to be associated to an unstable behavior of the dual variables caused by degeneracy of the master problem. The scope of this research, for the wireless sensor networks optimization, is to evaluate and compare a set of stabilization and intensification strategies, in order to identify their effect on the behavior of the dual variables, the number of iterations required for the column generation algorithm and finally on the computational time required to solve the problem. The results show that both strategies succeed to accelerate the proposed column generation approach. Furthermore, numerical evaluations indicate that intensification strategies might be useful to tackle the unstable behavior of the dual variables and to accelerate the solution process through column generation.

3.1 Introduction

A Wireless Sensor Network (WSN) is a net made of a large amount of battery-powered wireless sensors that are deployed to accomplish a set of monitoring and communication tasks. Two main considerations on WSN design are the energy usage and the coverage. Unfortunately, these two objectives may be contradictory. When a WSN is deployed to monitor a set of discrete targets, *e.g.*, to guarantee a minimum level of coverage, network lifetime can be considered as the time until WSN cannot satisfy that constraint. By contrast, if it is permitted to undercover the set of targets, network lifetime might be imposed as an operational constraint. As a consequence, a lower level of coverage may be necessary to guarantee that lifetime achieves the desired duration.

Some applications require the WSN to operate in remote or hostile environments. Consequently, a predefined location of sensors might not be possible, neither the replacement of batteries during network lifetime. When this is the case, the control of sensors' energy must be done *in situ*, *i.e.*, after sensors are actually deployed and net-

work topology is known. Thus, to extend the network lifetime, it might be necessary to adopt strategies to organize the use of the sensors in these uncertain conditions. In this way, it can be possible to improve the efficiency in the usage of sensors' battery budget while appropriate coverage of the interesting phenomena is provided.

A sensor is *active* if it is performing monitoring, processing, etc. tasks. If not, it is in an *idle* state in which it operates at negligible energy consumption rate and does not perform any activity within the network. In order to extend network lifetime, a typical approach is to deploy more sensors than actually needed. Then, it is possible to extend network lifetime by allocating the sensors to different subsets (covers). In this way, network lifetime can be extended by activating the covers (the sensors that belong to them) sequentially to monitor the targets assigning a time interval during which they are used that respects the battery capacity of each sensor [Deschinkel, 2011, Gentili and Raiconi, 2013].

WSN networks can be bandwidth constrained, *e.g.*, the number of channels used to transmit the information is limited and some sensors can not have channel access for data transmission [Cheng et al., 2005, Wang et al., 2009]. In this case, full target coverage can no longer be used as an indicator of a network properly operating. The Minimum Coverage Breach problem under Bandwidth constraints (MCBB) can be defined as the problem of generating and scheduling a set of covers guaranteeing a minimal network lifetime T_0 with the purpose of minimizing the total breach coverage, *i.e.*, the time-weighted sum of the uncovered targets along network lifetime. The network is composed of n homogeneous sensors, with the same battery autonomy and the same coverage radio R_s . The purpose is to minimize the coverage breach, the time that all targets are uncovered while the number of simultaneous active sensors is restricted to a bandwidth limit W .

Column generation (CG) algorithm has been proven to be a natural and efficient strategy to solve coverage and scheduling problems in WSN [Castaño et al., 2014, Deschinkel, 2011, Gentili and Raiconi, 2013, Gu et al., 2011, Rossi et al., 2012a]. The method iterates between a Restricted Master Problem (RMP) and a Pricing Subproblem (PS). RMP selects among the existent covers, the ones that allow to minimize the coverage breach, and the PS seeks for interesting covers that might be useful to improve the current objective function. In this chapter a decomposition based approach based on CG is adopted to solve MCBB. The method divides the problem in two. First, the RMP containing a reduced set of columns is used to allocate the optimal time intervals that each subset has to be used. Then, PS is used to identify additional attractive subsets of sensors that might be useful to improve the objective function of RMP. Each column contains the information about the structure of the subset sensors that compound the cover, *i.e.*, a column represents a set of sensors (or the corresponding energy consumption rates for those sensors) that could be scheduled using RMP to accomplish the monitoring tasks.

It is well known that CG often suffers from convergence issues that may limit its usability [Amor et al., 2009, Moun gla et al., 2010]. Gu *et. al.* [Gu et al., 2009b] report that, in a WSN application, the use of CG may keep on iterating with only marginal improvements on the objective function. Consequently, it may be necessary

the consideration of an approximated criterion to stop the optimization process, which may imply to accept a suboptimal solution. This problem is particularly notorious when the problem instances show degeneracy signals leading to an unstable behavior of the dual variables along CG iterations [Briant et al., 2008]. Extensive numerical experiments demonstrate that this drawback is present when the CG approach is applied to solve MCBB and other related WSN lifetime optimization problems. Then, the use of strategies to accelerate the convergence and to reduce the computational effort required to find an optimal solution is studied.

This chapter is structured as follows. Section 3.2 introduces the definition of MCBB, the approach based on column generation adopted to solve this problem and some highlights about the performance of this method. A set of strategies used to accelerate the CG and evaluations of their performance are presented in Section 3.3. Finally, the conclusions of this study and the future paths of research are summarized in section 3.4.

3.2 Problem description and related work

Consider a set $\mathcal{K} = \{k_1, k_2, \dots, k_m\}$ of targets with known locations and a set $\mathcal{S} = \{s_1, s_2, s_3, \dots, s_n\}$ of sensors deployed to cover the targets that are located within their sensing ranges R_s , *i.e.*, the maximum distance within which a sensor can be used to monitor a target. The maximum number of sensors that could be active at the same time is limited by the network bandwidth, W . The set $\Omega = \{C_1, C_2, \dots, C_l\}$ denotes the set of all the possible subsets of \mathcal{S} , such that $|C_j| \leq W$. A parameter b_j indicates the number of uncovered targets when cover C_j is used. The decision variable t_j is introduced to identify the time interval allocated to a feasible cover C_j . The minimum coverage breach problem under bandwidth constraints is to find a collection of pairs (C_j, t_j) that minimizes the breach coverage of the network, $\sum_{j|C_j \in \Omega' \subseteq \Omega} b_j t_j$, and guarantees a minimum lifetime T_0 .

In CG a cover C_j is represented by a column vector in RMP, which initially contains only a reduced set of columns $\Omega' \subseteq \Omega$. Then, RMP is solved to proven optimality and the dual variables values found are used as an input for PS. Next, PS is solved iteratively to identify interesting columns useful to reduce the total breach coverage. If a new column is found, this is added to RMP and the process starts again. Otherwise, the process stops. If the method used to solve PS is exact, it indicates that the current solution is optimal. The next section introduces the mathematical details of the decomposition approach used to solve MCBB and provides an analysis of the convergence of CG when it is applied to MCBB.

3.2.1 Mathematical approach

As previously mentioned, the solution approach adopted to solve MCBB divides the problem into time related and subset allocation decisions. The method guarantees that the battery capacity of each sensor is respected by imposing individual constraints for them in terms of the total power consumed during the network lifetime. In the

following, the decomposition based approach used to find optimal solutions for MCBB is presented.

3.2.1.1 Master problem

We assume that all the sensors are homogeneous and consume energy at the same rate when they are active. The parameter a_{ij} denotes the energy consumption rate of sensor s_i if it is included in the cover set C_j . When the sensor is idle it takes the value 0. The RMP used to solve the MCBB is presented as follows [Rossi et al., 2012a]:

Minimize :

$$\sum_{j|C_j \in \Omega'} b_j t_j \quad (3.1)$$

Subject to:

$$\sum_{j|C_j \in \Omega'} a_{ij} t_j \leq 1 \quad \forall i \in \mathcal{S} \quad (3.2)$$

$$\sum_{j|C_j \in \Omega'} t_j \geq T_0 \quad (3.3)$$

$$t_j \geq 0 \quad \forall j|C_j \in \Omega' \quad (3.4)$$

As mentioned above, the masters' problem objective (3.1) is to minimize the coverage breach. The set of constraints (3.2) guarantee that a sensor is only used for a maximum time bounded by the battery lifetime (normalized to 1). The purpose of constraint (3.3) is to guarantee that the minimum network lifetime is reached by using the selected covers. Note that the time that a cover set is used is continuous and is not bounded directly for the model; however, it is bounded indirectly by the battery lifetime constraints (3.2).

3.2.1.2 Pricing subproblem

PS is used as column generator. To generate a new cover set C_j an iteration of CG is performed, and the corresponding interesting column j is added to RMP. For each column, a new decision variable t_j is created. This variable has associated the coefficients a_{ij} in RMP. Let L_k be the set of sensors able to cover the target k and $r_k \in \{0, 1\}$ an auxiliary variable introduced to indicate whether or not a target k is covered by the active set of sensors. We denote by π_i the cost of an active sensor, obtained as the dual variable value associated with the sensors battery constraints (3.2), and Π the dual variable associated with the minimum lifetime constraint (3.3). In this way, the cover structure $C_j = [a_{1,j}, a_{2,j}, \dots, a_{n,j}, 1]$, where the 1 is related with the minimum lifetime constraint (3.3), is defined through the PS as follows:

Minimize :

$$b_j - \sum_{i \in \mathcal{S}} a_{ij} \pi_i - \Pi \quad (3.5)$$

Subject to:

$$\sum_{i \in \mathcal{S}} a_{ij} \leq W \quad (3.6)$$

$$b_j = |\mathcal{K}| - \sum_{k \in \mathcal{K}} r_k \quad (3.7)$$

$$r_k \leq \sum_{i \in L_k} a_{ij} \quad \forall k \in \mathcal{K} \quad (3.8)$$

$$r_k \in \{0, 1\} \quad \forall k \in \mathcal{K} \quad (3.9)$$

$$a_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{S} \quad (3.10)$$

The purpose of PS is to define a cover structure with minimum reduced cost (3.5). Equation (3.6) is used to guarantee that the bandwidth limit is not exceeded. Constraint (3.7) defines the breach coverage used to calculate the reduced costs and then used as the cost of the generated column in RMP. Sets of constraints (3.8) are used to help the model to define the coverage breach and the covering status of a target.

3.2.1.3 Convergence of the Column Generation algorithm

It is widely reported in the literature that, in many applications, the CG shows convergence issues [Lubbecke and Desrosiers, 2005, Westerlund, 2005]. Figure 3.1 presents the typical evolution of the objective function of MCBB along iterations of CG. It is possible to observe that while a near-optimal solution is approached considerably fast, the improvement of the objective function is slow in the last iterations. This phenomenon, known as *tail-off* effect, has been widely studied and is reported to be present in pathological cases of slow convergence in CG [Briant et al., 2008, Gilmore and Gomory, 1961, 1963].

Several explanations for the tail-off have been proposed; however, most authors agree that this effect is strongly associated to an unstable behavior of the dual variables, *i.e.*, in consecutive iterations the values taken for the dual variables are very far from the previous ones [Briant et al., 2008]. Moreover, the distance $\|\pi^j - \pi^*\|$, where π^* is the optimal dual solution, does not necessarily decrease along consecutive iterations (see Figure 3.2), *i.e.*, $\|\pi^j - \pi^*\| < \|\pi^{j+1} - \pi^*\|$ [Vanderbeck, 2005]. In addition, the presence of alternative solutions for PS implies that the column returned could be basically selected at random among the pool of optimal solutions without considering CG convergence. That means that the CG algorithm is almost completely unable to exploit the fact that it has already reached a good set of dual variables values, near to the optimal dual solution, to improve the overall convergence [Amor et al., 2009].

Dual variables stabilization (DVS) [Du Merle et al., 1999, Marsten et al., 1975] is proposed as a strategy to avoid the large oscillations of the dual variables values by imposing a set of bounds to the values they can take. If a good dual estimation is available, the dual values could be restricted to a small region around the optimal solution in such a way that PS only produces near optimal columns which could accelerate the overall CG convergence [Amor et al., 2009]. Unfortunately, most of the time this is not

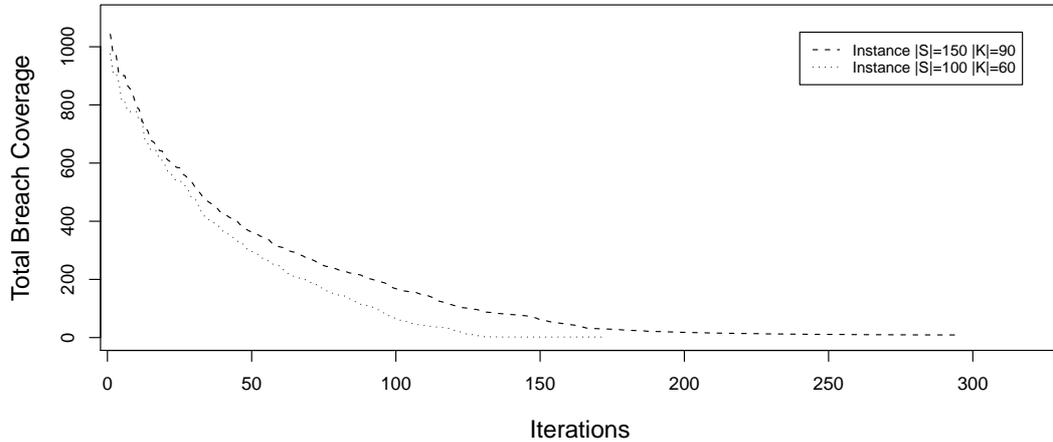


Figure 3.1 – Tail-of effect.

the case. So, the idea behind DVS is to keep under control the dual variables values at each iteration and try to guide them in a smooth fashion to the optimal dual solution.

In order to manage the unstable behavior of the dual variables, Elhallaoui *et al.* [Elhallaoui et al., 2008] propose the dynamic constraint aggregation (DCA) to be used in highly degenerate set partitioning problems. The basic idea of this method is to reduce the solution space by dealing dynamically with aggregated versions of the master problem. In this way, the effects of degeneracy are expected to decrease as a consequence of the reduction on the number of basis generating degenerate pivots. Benchimol *et al.* [Benchimol et al., 2012] extended the DCA method by applying DVS strategies. The result obtained shows that the proposed combination highly outperforms the effect of the individual methods in set partitioning problems.

On the other hand, intensification procedures have been proposed also as interesting strategies to improve the performance of CG [Desrochers et al., 354]. In this latter approach, the idea consists in returning several columns, instead of one, at each iteration of CG using different criteria to select the ones that are interesting. Macambira *et al.* [Macambira et al., 2005] and Touati *et al.* [Moungla et al., 2010] report the benefits obtained by this approach. They report a decrease on the total number of iterations required to reach an optimal solution.

The use of intensification strategies may lead to a faster increase in the size of RMP. Moreover, most columns included are useless because they will not be part in the optimal solution. As a consequence, strategies to control these undesirable effects might be required. In this way, it is possible to control the enlargement of RMP size whereas it is improved the performance of CG to solve the problems it tackles. Touati *et al.* [Moungla et al., 2010], propose the use of κ -intensification and diversification approaches to accelerate the CG convergence without overly increasing the size of the

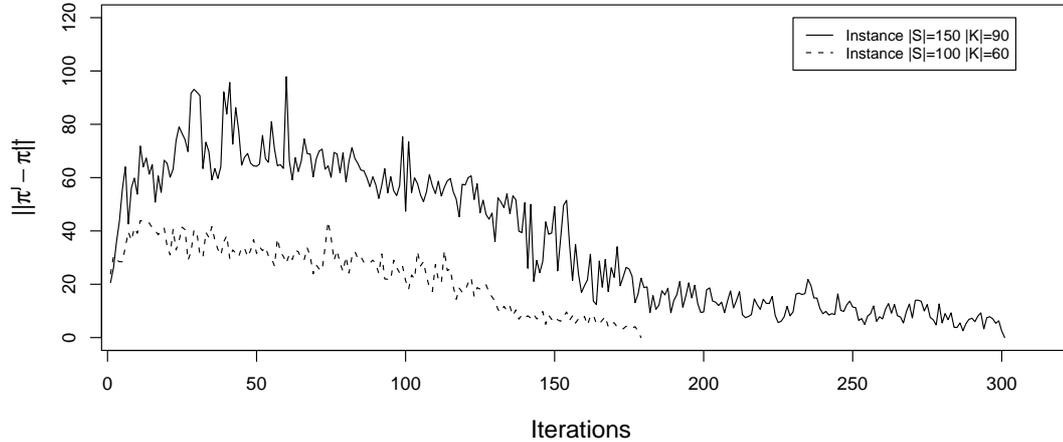


Figure 3.2 – Unstable behavior of dual variables along successive CG iterations.

RMP. However, such a filter might be not necessary if the RMP is not too complicated to solve. A description of these approaches will be provided in Section 3.3.2.

In this chapter we evaluate a set of stabilization and intensification procedures to identify the most suitable to be used in MCBB. The methods are evaluated in terms of convergence on MCBB measured as the number of iterations required to reach an optimal solution. Furthermore, the effect of both strategies to reduce the unstable behavior of dual variables strategies is presented. To improve performance, we provide an intensification strategy by using a genetic algorithm which is used to solve the PS within the CG framework. Nevertheless, when the genetic algorithm fails to find a new interesting column, it is followed by an exact method which allows to verify whether or not the current solution is optimal.

3.3 Acceleration strategies for the column generation algorithm

In this section, we evaluate the use of stabilization and intensification strategies to improve the convergence of the CG framework when applied to the coverage and scheduling problems in wireless sensor networks, particularly (MCBB). Both methods are evaluated in terms of their suitability to accelerate CG. Furthermore, the effect on the evolution of dual variables is evaluated as well as the technical specifications required to make a successful implementation of both strategies.

3.3.1 Dual variables stabilization

The purpose of stabilization methods is to use the information offered by the dual variables to keep a smooth convergence to the optimal dual solution accelerating the CG. In dual-variables-based stabilization (DVS), the strategy is to manage the information of the dual variables by controlling the changes of their values to produce interesting columns around the optimal solution. Two main strategies are discussed in this chapter, the BoxStep based methods comprising the Marsten's method [Marsten et al., 1975] and its generalization proposed by Du Merle *et al.* [Du Merle et al., 1999]. A second type of strategy presented by Neame proposes to generate columns using a linear combination of the dual variables obtained in the previous iterations [Neame, 2000]. Although simple, this method used to show interesting practical results. A description of these methods will be presented in the next sections.

3.3.1.1 BoxStep stabilization method

Marsten *et al.* [Marsten et al., 1975] proposed a technique to accelerate CG convergence and stabilize the behavior of dual variables. The method works on the dual problem by defining a set of bounds intended to avoid the dual variables to move too far from their current values. This method, widely known as *BoxStep*, requires the dual variables to take values within the box.

Consider the following dual problem associated to the model (3.1)-(3.4):

Maximize :

$$\sum_{i \in \mathcal{S}} \pi_i + \Pi T_0 \quad (3.11)$$

Subject to:

$$\sum_{i \in \mathcal{S}} a_{ij} \pi_i + \Pi \leq b_j \quad \forall j | C_j \in \Omega' \quad (3.12)$$

$$\pi_i \leq 0 \quad \forall i \in \mathcal{S} \quad (3.13)$$

$$\Pi \geq 0 \quad (3.14)$$

The idea of BoxStep method, is to impose upper bounds δ_i, Δ^+ and lower bounds δ_i^-, Δ^- on the dual variables values associated to RMP constraints in order to limit the values that they can take in an iteration of CG. In order to adapt the Marsten's method, the next set of constraints must be added to the dual problem (3.11)-(3.14):

$$\pi_i \geq \delta_i^- \quad \forall i \in \mathcal{S} \quad (3.15)$$

$$\pi_i \leq \delta_i^+ \quad \forall i \in \mathcal{S} \quad (3.16)$$

$$\Pi \geq \Delta^- \quad (3.17)$$

$$\Pi \leq \Delta^+ \quad (3.18)$$

Constraints (3.15) and (3.17) impose a lower bound on the dual variables values associated with the battery lifetime constraints and the minimum network lifetime respectively. In the same way, the constraints (3.16) and (3.18) are used to impose upper bounds on the dual variables. The overall effect of this strategy is to keep the dual variables value inside a box centered at the middle of the range $[\delta_i^-, \delta_i^+]$.

In RMP, the addition of the dual variables bounds implies the raise of a new set of primal variables, y_i^+, y_i^-, Y^+ and Y^- , with a penalty corresponding to the dual variables bounds. This means that CG is performed on relaxed versions of RMP, so it is possible to take infeasible solutions for the problem along the CG iterations.

Minimize :

$$\sum_{j|C_j \in \Omega'} b_j t_j + \sum_{i \in \mathcal{S}} \delta_i^+ y_i^+ - \sum_{i \in \mathcal{S}} \delta_i^- y_i^- + \Delta^+ Y^+ - \Delta^- Y^- \quad (3.19)$$

Subject to:

$$\sum_{j|C_j \in \Omega'} a_{ij} t_j + y_i^+ - y_i^- \leq 1 \quad \forall i \in \mathcal{S} \quad (3.20)$$

$$\sum_{j|C_j \in \Omega'} t_j + Y^+ - Y^- \geq T_0 \quad (3.21)$$

$$t_j \geq 0 \quad \forall j|C_j \in \Omega' \quad (3.22)$$

$$y_i^+ \geq 0, y_i^- \geq 0 \quad \forall i \in \mathcal{S}, \quad Y^+, Y^- \geq 0 \quad (3.23)$$

In order to guarantee a valid solution at the end of the CG algorithm, it is necessary to verify that none of these variables appear in the final solution. Once an optimal solution is found, the values of the "relaxation variables" are evaluated. If the solution is not feasible for the original problem, the penalty for these variables is increased. As a consequence, the "Box Sizes" are increased. The process is repeated until the optimal solution for PS is non-negative, *i.e.*, $b_j - \sum_{i \in \mathcal{S}} a_{ij} \pi_i - \Pi \geq 0$, which means that there are no other columns to improve the optimal solution.

The method, however, has a drawback. It allows a lot of flexibility with regard to parameter selection and no clear rules are established to improve its performance [Amor et al., 2009, Tran et al., 2006]. Consequently, a deep knowledge of the model details and instance's characteristics might be required to set up a successful implementation leading to accelerate the convergence of CG. Three mechanisms are considered in this research to update the box center: *Stationary*, *sliding* and *hybrid* [Tran et al., 2006, Westerlund, 2005]. Let μ_i^j be the box center associated to the dual variable π_i in the BoxStep process and B_j the box size in an iteration j , that is $B_j = \delta_i^+ - \delta_i^-$, where the values $\delta_i^+ = \mu_i^j + \frac{B_j}{2}$ and $\delta_i^- = \mu_i^j - \frac{B_j}{2}$ denote the dual variable bounds. A brief description of the methods is presented below:

Stationary Center In this updating strategy the center μ_i^j is the same during the whole CG process, *i.e.*, $\mu_i^{j+1} = \mu_i^j = \mu_i^0$, where μ_i^0 is the center used at the beginning of CG process.

Sliding Center At each iteration of the CG, the box centers are updated. In this case, the values for the centers take the value of the current dual solution, it is $\mu_i^{j+1} = \pi_i$.

Hybrid Center In this strategy, the centers are updated only when the dual variables do not present any slack regarding the box bounds in an iteration of CG.

For the three updating methods it is possible to observe that, if no new column can be found, and the set of variables y_i^+, Y^+ and y_i^-, Y^- have values greater than zero, the solution is not yet feasible for the RMP and the bounds need to be updated. This means that the value of B_j is enlarged and the primal penalty increased. In that case, the box size is enlarged by $x\%$, $B_j = (1 + x)B_j$.

3.3.1.2 Generalized BoxStep method

Du Merle *et al.* [Du Merle et al., 1999] have proposed a generalization of the BoxStep method. In this case, a three-piecewise function allows the dual variables to take values outside of the box if a penalty is incurred. In terms of RMP the original problem is modified in such a way that the set of variables y_i^+, y_i^- are obligated to remain below a certain value ϵ_i , the penalty term in the associated dual problem. Let γ_i^+ and γ_i^- be the dual variables associated with the relaxation of the dual bounds respectively, the dual problem for MCBB under Du Merle's method could be expressed as follows:

Maximize :

$$\sum_{i \in S} \pi_i + \Pi T_0 + \sum_{i \in S} \epsilon_i^- \gamma_i^- + \sum_{i \in S} \epsilon_i^+ \gamma_i^+ + \zeta^- \omega^- + \zeta^+ \omega^+ \quad (3.24)$$

Subject to:

$$\sum_{i \in S} a_{ij} \pi_i + \Pi \leq b_j \quad \forall j | C_j \in \Omega' \quad (3.25)$$

$$\pi_i + \gamma_i^+ \leq \delta_i^+ \quad \forall i \in S \quad (3.26)$$

$$-\pi_i + \gamma_i^- \leq -\delta_i^- \quad \forall i \in S \quad (3.27)$$

$$\Pi + \omega^+ \leq \Delta^+ \quad (3.28)$$

$$-\Pi + \omega^- \leq -\Delta^- \quad (3.29)$$

$$\pi_i, \gamma_i^+, \gamma_i^- \leq 0 \quad \forall i \in S \quad (3.30)$$

$$\omega^+, \omega^- \leq 0 \quad (3.31)$$

$$\Pi \geq 0 \quad (3.32)$$

It is possible to identify three regions for each variable in the model above. First, a region with no penalty in which it is possible to satisfy the constraints (3.26)-(3.29) without incurring in any penalization (for the use of the variables $\gamma_i^-, \gamma_i^+, \omega^+, \omega^-$). The other two regions are used to allow a dual variable to take values out of these bounds.

The counterpart is to pay a penalty on the objective function that increases linearly with the violation.

From the viewpoint of RMP, the original model is transformed as follows:

Minimize :

$$\sum_{j|C_j \in \Omega'} b_j t_j + \sum_{i \in S} \delta_i^+ y_i^+ - \sum_{i \in S} \delta_i^- y_i^- + \Delta^+ Y^+ - \Delta^- Y^- \quad (3.33)$$

Subject to:

$$\sum_{j|C_j \in \Omega'} a_{ij} t_j + y_i^+ - y_i^- \leq 1 \quad \forall i \in S \quad (3.34)$$

$$\sum_{j|C_j \in \Omega'} t_j + Y^+ - Y^- \geq T_0 \quad (3.35)$$

$$y_i^+ \leq \epsilon_i^+ \quad \forall i \in S \quad (3.36)$$

$$y_i^- \leq \epsilon_i^- \quad \forall i \in S \quad (3.37)$$

$$Y^+ \leq \zeta^+ \quad \forall i \in S \quad (3.38)$$

$$Y^- \leq \zeta^- \quad \forall i \in S \quad (3.39)$$

$$y_i^+, y_i^- \geq 0 \quad \forall i \in S \quad Y^+, Y^- \geq 0 \quad (3.40)$$

$$t_j \geq 0 \quad \forall j|C_j \in \Omega' \quad (3.41)$$

It is possible to observe that, either for Marsten's BoxStep and Du Merle's methods, it is required the initialization and updating of the box centers μ_i^j , box width B_j and dual penalties $\epsilon_i^+, \epsilon_i^-, \zeta^+, \zeta^-$. This implies that a lot of tuning work could be required to select a good set of parameters. Furthermore, a good choice of the parameters value also may rely on the characteristics of each problem instance.

As in the Marsten's BoxStep method, the Du Merle's method allows a lot of flexibility regarding the selection of the initial parameters and updating strategies. The same set of strategies used in the BoxStep method of section 3.3.1.1 is applied and evaluated to improve the performance. In addition, the dual penalty terms $\epsilon_i^+, \epsilon_i^-, \zeta^+$ and ζ^- are initialized with the right hand side values of the associated constraint in the original model [Du Merle et al., 1999], *i.e.*, equal to 1. In order to reduce their values, a factor df is defined, then after each iteration a reduction equal to $(1 - df)\%$ is applied to the dual penalty terms.

3.3.1.3 Neame's stabilization method

Neame [Neame, 2000] proposes an algorithm to accelerate the CG without using neither stabilizing terms nor including additional variables in the original model. The method proposes to use a linear combination of the current dual variables values and the ones obtained at previous iterations to solve PS and find a new interesting column. Let π^k be the set of dual values of the primal problem at iteration k , and θ_k the dual values obtained as a linear combination and used for PS. At iteration k the values for θ_k are updated in the following way:

$$\theta_k = \alpha\pi^k + (1 - \alpha)\theta_{k-1} \quad (3.42)$$

where $\alpha \geq 0$ is a parameter.

New columns are profitable only if they have an interesting reduced cost based on the values π^k . When non-profitable reduced cost column is found (*missprice*), a new iteration of CG is performed using only these original dual values for the current CG iteration PS. The method finishes when no profitable columns exists and current solution is optimal.

3.3.1.4 Dual variable values initialization

It is well known that the selection of initial dual box centers could have a significant effect on the number of CG iterations. Several strategies have been proposed to select initial dual values. Du Merle *et al.* [Du Merle et al., 1999] and Ben Amor *et al.* [Amor et al., 2009] propose to approach the optimal solution using heuristic methods and use the dual variable values to initialize the stabilized CG process. As a default, when no information is available about the optimal dual solution, it is permissible to use $\mu_i = 0$. Nonetheless, in general this is not a good choice and previous experiences show that the performance of stabilization approaches could be highly improved by selecting a good set of initial box centers. An incorrect choice, by contrast, can be counterproductive to the stabilizing effect and so enlarges the CG process .

In order to initialize the box center of the dual variables in the stabilization process, an initial solution is approached through a greedy heuristic. The dual value corresponding to this initial solution is used as the initial center in the dual variable stabilization. In the same way, the initial solution obtained is used as the initial basis in the CG process.

3.3.1.5 Computational experiments

The purpose of this section is to evaluate stabilization strategies. The effects of dual stabilization on the dual variables are presented as well as the computational benefits of using such strategies. In the same way, the updating strategies are compared for the BoxStep based methods in order to identify the best strategy for MCBB. Finally, the effect of the value α on the performance of the Neame's method is evaluated for the case of convex combination of previous dual variables values.

The proposed approaches are implemented in Python and executed on an Intel Core i-5 processor at 1.6 GHz with 2 GB of RAM running under OS-X Lion. The version 4.0.2 of Gurobi optimization engine is used to solve RMP and PS. Eight classes of instances are considered for analysis. They are defined by the number of sensors n , the number of targets to be covered $|K|$ and the sensing range R_S . As proposed by Wang *et al.* [Wang et al., 2009] and Rossi *et al.* [Rossi et al., 2012a] it is assumed that sensor and targets are randomly deployed in a 500x500 square area. For each category a group of three instances is designed and experiments are performed for two values of bandwidth, $W = 5$ and $W = 10$.

Several authors have devoted efforts to evaluate the impact of the parameters selected for the initial box sizes B_j and center μ_i , the percentage x of enlargement of the box size and the initial dual penalty for the Du Merle's method [Amor et al., 2009, Westerlund et al., 2006]. In order to perform standard experiments, a common set of parameters is selected based on the observations obtained through some preliminary experiments. The initial box size for the dual variables values is selected as 0.5 times the value of the right hand side (RHS) for the battery lifetime constraints (Eq. 3.2). The initial observations also show that unstable behavior was mainly caused by the dual variables values associated to the battery lifetime constraints with little affectation coming from the minimum network lifetime constraint (Eq. 3.3). For this reason, it is proposed the use of big size initial boxes for the dual variable Π associated to this constraint, in this way is avoided any possible bias introduced in the evolution of this variable along CG iterations.

Figure 3.3 shows a comparison of the evolution of the quantity $\|\pi^j - \pi^*\|$ along the iterations j of CG. It can be seen that the use of BoxStep method and the generalization presented by Du Merle (CGS3P) reduces considerably the oscillations when using the CG without stabilization. In addition, it is possible to observe that, in general, the use of stabilization techniques based on box penalties produces a smooth convergence to the optimal dual solution. As presented in Figure 3.3, by using DVS a step effect appears. This effect is related with the box sizes updates (see Section 3.3.1.1). Nevertheless, most of the time a smooth behavior appears between each pair of steps as a consequence of the application of stabilization strategies.

The three updating strategies of stabilization methods, namely hybrid, sliding and stationary, are compared in terms of computational time and average number of iterations or equivalently the number of columns generated. Tables 3.1 and 3.2 present this comparison. The column headers n , $|K|$, R_s and W indicate the instances characteristics. Each line presents the average obtained by using the approach on three randomly generated instances with the same specifications.

The average variation on the number of iterations compared to CG without stabilization is also presented. The results show that most stabilization strategies lead to reductions on the number of iterations compared to the CG without stabilization. However, the stationary updating strategy does not provide a significant advantage in terms of convergence compared to the CG without stabilization. In this study, the same values for initial box size and percentage of enlargement are used for similar instances groups in order to analyze the overall performance of the method more than the best

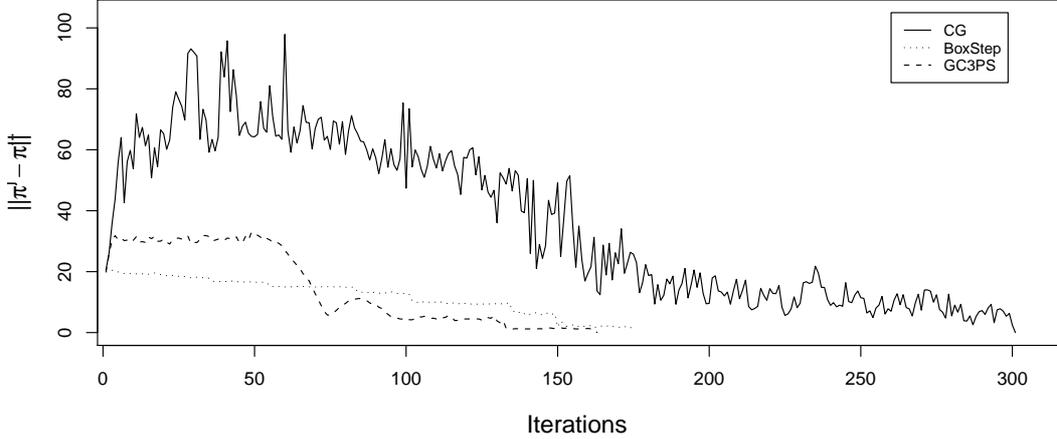


Figure 3.3 – Evolution of $\|\pi^j - \pi^*\|$ along the iterations using BoxStep method (BoxStep) and du Merle's method (GC3PS).

case performance for each instance.

n	K	R_s	W	Hybrid		Sliding		Stationary		No stabilization		
				Time (s)	#Cols	Time (s)	#Cols	Time (s)	#Cols	Time (s)	#Cols	
50	30	130	5	1.86	90.00	1.77	87.00	2.33	78.00	2.06	99.67	
			10	0.52	31.33	0.55	31.67	0.71	40.67	0.87	52.33	
			150	5	1.64	78.00	1.49	66.67	1.70	82.00	1.61	80.00
				10	0.54	26.67	0.52	26.00	0.49	25.00	0.77	39.33
100	60	130	5	47.99	253.00	51.82	238.33	152.10	282.00	41.87	304.33	
			10	8.04	121.00	8.42	122.67	8.25	135.67	10.30	142.00	
			150	5	27.06	232.00	30.60	200.00	93.71	262.33	28.08	249.67
				10	6.73	84.67	6.63	84.00	6.69	92.67	7.51	99.67
150	90	130	5	242.24	432.33	361.12	417.33	877.64	544.33	234.34	468.67	
			10	29.64	200.00	30.75	205.00	25.88	207.67	37.35	264.00	
			150	5	215.40	405.00	398.97	379.00	857.28	530.67	250.86	441.67
				10	23.76	128.67	22.90	124.33	20.88	123.33	23.73	134.67
200	120	130	5	3224.89	761.67	4681.22	709.33	2630.60	843.33	3518.33	879.67	
			10	100.94	301.67	88.03	292.33	76.88	289.33	102.29	365.00	
			150	5	2345.87	605.33	2050.08	582.00	2563.28	652.00	2283.45	653.00
				10	59.44	160.00	55.96	147.00	43.15	146.33	61.19	181.33
Average variation				-9%	-15%	0%	-19%	52%	-8%	-	-	

Table 3.1 – Comparison of the updating strategies in the BoxStep method.

With a few exceptions, the strategies are able to reduce the efforts devoted to solve MCBP reducing the computational time and the number of iterations. However, as presented in Tables 3.1 and 3.2, the reductions in computational time are not proportional to the reductions on the number of iterations. The experiments show that in most cases a reduction in computational time and number of iterations is obtained. The best performance is shown by the hybrid updating strategy of section 3.3.1.1. However, even in this case, the ratio between the average percentual reduction in number of iterations

	K	R_s	W	Hybrid		Sliding		Stationary		No stabilization			
				Time (s)	#Cols	Time (s)	#Cols	Time (s)	#Cols	Time (s)	#Cols		
50	30	130	5	1.62	57.00	1.53	58.33	1.50	59.67	2.06	99.67		
			10	0.79	22.00	0.77	20.67	0.90	27.33	0.87	52.33		
		150	5	1.32	41.33	1.29	42.33	1.27	47.00	1.61	80.00		
			10	0.32	4.00	0.42	7.33	0.42	8.33	0.77	39.33		
100	60	130	5	38.24	167.67	45.25	173.67	44.34	205.67	41.87	304.33		
			10	8.15	79.00	8.52	83.33	8.87	95.33	10.30	142.00		
		150	5	23.24	145.33	22.96	153.67	24.20	172.00	28.08	249.67		
			10	3.66	25.33	3.39	22.00	3.49	23.00	7.51	99.67		
		150	90	130	5	266.86	319.67	300.92	339.00	212.70	346.33	234.34	468.67
					10	33.58	126.33	38.10	138.33	43.44	170.00	37.35	264.00
150	5			309.74	301.33	378.57	333.33	230.01	335.67	250.86	441.67		
		10	12.83	39.33	12.64	39.00	12.23	35.67	23.73	134.67			
200	120	130	5	3291.33	599.00	4277.14	635.33	3051.96	645.67	3518.33	879.67		
			10	150.07	203.67	148.01	214.33	149.95	251.33	102.29	365.00		
		150	5	2908.94	477.67	3222.84	499.33	2185.13	485.33	2283.45	653.00		
			10	45.87	53.33	42.39	45.67	38.37	55.33	61.19	181.33		
Average variation				-11%	-50%	-5%	-48%	-14%	-43%	-	-		

Table 3.2 – Comparison of updating strategies in the Du Merle’s method.

and computational time is 5:1. This indicates that the time required to perform each iteration in the methods that use dual variable stabilization are higher than in CG without stabilization.

As for Neame’s method, the results confirm that it is always useful to reduce the number of iterations required to reach an optimal solution. Table 3.3 presents a comparison among the average number of iterations and computational time obtained through Neame’s method and CG without stabilization. The results show that this method is useful to accelerate the convergence of CG. The method is tested in the same set of instances as the BoxStep based methods using an arbitrary α value of 0.5. Furthermore, it is observed an average reduction of 42% in the number of iterations and 35% in computational time compared with the CG without stabilization. Additionally, the Neame’s method generally produces a smoothing effect on the evolution of the dual variables, measured by the distance to the optimal dual solution $\|\pi^j - \pi^*\|$, even though this is not directly the purpose of this method (see Figure 3.4).

Tables 4 and 5 present a comparison among the number of iterations and computational time, respectively, obtained for a selected group of randomly generated instances and α values in the range 0.1-1.0. The columns labeled with α in the range 0.1-0.9 correspond to the values obtained by using the Neame’s method in the number of iterations and the computational time respectively. By contrast, the last column corresponds to the number of iterations without stabilization ($\alpha = 1$). It is observed that Neame’s method outperforms the CG without stabilization with a reduction of about 21% on the average number of iterations and 11% in computational time. That means that whereas a smaller number of iterations is required, the time required to perform an iteration is higher than in the CG without stabilization. In the tables, the bold numbers are used to highlight the best value obtained with the different α possibilities. Although the parameter has a significant impact on the performance of the method, there is no single value better than the others to reduce the number of iterations. Nevertheless, small values of α seems to perform better when $W = 5$.

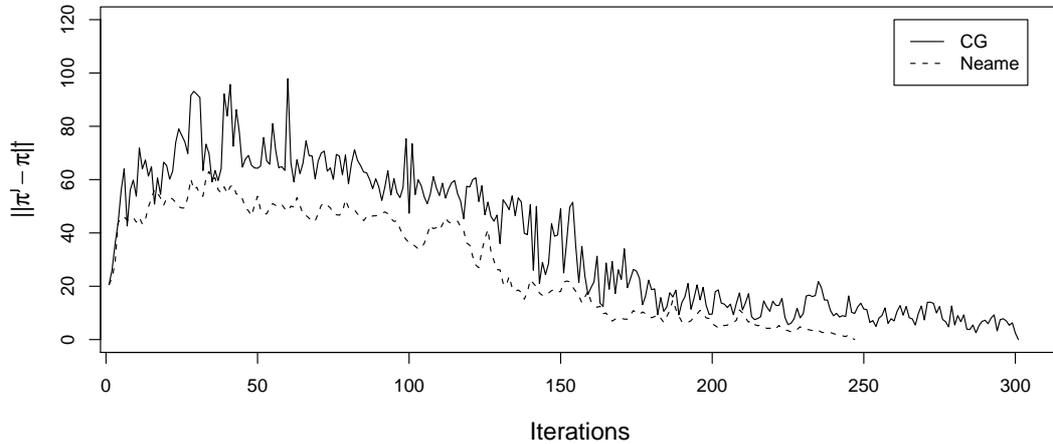


Figure 3.4 – Comparison of dual variables behavior without stabilization (CG) and the Neame’s method.

In general, the results obtained through the application of stabilizing strategies within the CG framework used to solve MCBB seem to indicate that the method proposed by Neame outperforms the DVS strategies. By using this approach it is possible to obtain interesting reductions regarding to computational time and number of iterations. The experimental results show that this strategy was always useful to reduce the computational time required to solve MCBB. Furthermore, this method is simple enough to avoid requiring a large computational effort. The results seem to indicate that by using this approach it is possible to obtain reductions in computational time compared to DVS studied here.

3.3.2 Intensification strategies

In order to evaluate the effect of the use of intensification strategies in MCBB, a set of experiments using this strategy are performed. Two methods are evaluated, κ -Intensification and Diversification. A description of these methods is presented below.

3.3.2.1 κ -Intensification

In a pure intensification strategy, it is possible to return to the RMP all the interesting columns found through PS to the RMP at each iteration of CG. This strategy usually leads to reduction in the number of iterations in CG; however, it could lead also to an overloaded RMP with a huge number of variables that make each iteration of CG slower. Touati *et al.* [Moungla et al., 2010] propose to limit the number of added columns to only those κ that are most interesting regarding to the defined criterion, typically the reduced cost. The purpose of such a strategy is to control the problem size taking

n	K	R _s	W	Neame		No stabilization		
				Time	#Cols	Time	#Cols	
50	30	130	5	0.86	54.33	2.06	99.67	
			10	0.66	26.00	0.87	52.33	
			150	5	1.02	41.33	1.61	80.00
			10	0.46	23.00	0.77	39.33	
100	60	130	5	9.32	129.33	41.87	304.33	
			10	6.30	92.00	10.30	142.00	
			150	5	18.96	143.00	28.08	249.67
			10	4.13	30.00	7.51	99.67	
150	90	130	5	180.39	352.33	234.34	468.67	
			10	31.46	118.00	37.35	264.00	
			150	5	376.57	343.67	250.86	441.67
			10	10.92	58.00	23.73	134.67	
200	120	130	5	2202.11	600.33	3518.33	879.67	
			10	48.87	215.67	102.29	365.00	
			150	5	2320.29	633.67	2283.45	653.00
			10	15.19	85.00	61.19	181.33	
Average variation				-35%	-42%	-	-	

Table 3.3 – Comparison of Neame’s method and CG without stabilization.

n	K	R _s	W	α										
				0, 1	0, 2	0, 3	0, 4	0, 5	0, 6	0, 7	0, 8	0, 9	1	
50	30	130	5	60	55	81	82	88	78	87	91	98	92	
			10	29	36	32	36	38	35	31	36	32	43	
			150	5	67	75	72	71	74	68	74	74	69	84
			10	27	25	26	24	25	24	25	24	22	40	
100	60	130	5	209	209	231	224	230	250	245	241	266	281	
			10	138	125	128	114	124	114	118	125	117	158	
			150	5	156	164	203	215	202	197	201	208	224	241
			10	92	85	90	82	76	65	67	64	67	93	
150	90	130	5	393	383	419	407	432	463	458	478	478	531	
			10	155	140	143	121	126	120	121	129	130	206	
			150	5	302	306	320	317	345	333	345	358	364	372
			10	126	105	100	93	85	83	70	78	80	120	
Average variation				-22%	-23%	-19%	-21%	-19%	-23%	-22%	-20%	-19%	-	

Table 3.4 – Evaluation of the effect of α in the number of iterations in the Neame’s method.

n	K	R _s	W	α										
				0, 1	0, 2	0, 3	0, 4	0, 5	0, 6	0, 7	0, 8	0, 9	1	
50	30	130	5	1, 11	0, 96	1, 64	1, 63	1, 76	1, 89	1, 77	1, 90	2, 13	1, 65	
			10	0, 46	0, 56	0, 49	0, 56	0, 59	0, 53	0, 50	0, 57	0, 51	0, 67	
			150	5	1, 08	1, 61	1, 49	1, 55	1, 41	1, 35	1, 41	1, 46	1, 31	1, 63
			10	0, 52	0, 48	0, 49	0, 46	0, 48	0, 45	0, 49	0, 47	0, 43	0, 74	
100	60	130	5	23, 98	71, 52	76, 12	62, 44	64, 37	66, 43	61, 03	65, 44	64, 30	56, 30	
			10	4, 08	7, 46	7, 77	6, 89	7, 52	6, 82	7, 15	7, 63	7, 10	9, 75	
			150	5	17, 99	23, 74	38, 12	29, 99	30, 45	29, 17	28, 99	26, 54	29, 31	26, 65
			10	7, 40	6, 43	6, 82	6, 15	5, 80	4, 93	5, 08	4, 88	5, 10	7, 06	
150	90	130	5	159, 71	545, 24	466, 68	440, 43	420, 98	463, 61	533, 74	386, 78	416, 03	363, 22	
			10	22, 39	20, 49	20, 98	17, 86	18, 66	17, 71	17, 89	19, 17	19, 28	30, 71	
			150	5	84, 47	152, 17	151, 37	146, 82	133, 85	108, 38	108, 79	121, 73	107, 02	112, 99
			10	24, 14	20, 18	19, 34	18, 02	16, 51	16, 13	13, 76	15, 26	15, 67	23, 54	
Average variation				-31%	-6%	0%	-8%	-8%	-12%	-12%	-12%	-12%		

Table 3.5 – Evaluation of the effect of α in the computational time (s) for the Neame’s method.

advantage of the benefits offered by the intensification strategy.

3.3.2.2 Diversification

In order to keep under control the RMP size and include diversification features in the returned columns, some authors recommend to add to RMP only columns contributing to different constraints [Desrochers et al., 354, Mougla et al., 2010]. Within the CG framework these strategies are known as diversification and two main strategies are distinguished by the authors, *diversification by resolution* and *diversification by selection*.

Diversification by resolution is especially useful when the optimal solution for PS is easy to obtain. It consists in computing at each iteration of CG an interesting set of columns that are 0-neighbors between them using as a criterion to generate them the reduced cost. In the context of MCBB, it computes iteratively new solutions for PS removing those sensors (imposing to them a fixed value of zero) that already belong to a previously generated cover. By using *diversification by selection* the idea is to return to RMP only the columns found during the PS solution process that do not contribute to the same constraints but are still feasible and useful in terms of the possible contribution to improve the objective function. This strategy is mostly used when the algorithms applied to solve PS are able to compute more than a single solution along the solution process for PS. Within MCBB, this strategy selects and returns at each iteration several covers based not only on the reduced cost criterion but also in how similar it is compared to the best solution found.

In this study, the proposal is to use diversification by selection to generate columns in a diversified manner and try to accelerate the CG convergence by avoiding the inclusion of columns with similar structure to the best one obtained based on the reduced cost criterion.

3.3.2.3 Intensification and diversification through a Genetic Algorithm

To include diversification and intensification we propose a GA to find approximated solutions for PS. Two strategies are considered. First, a raw group of the k best columns found will be added to RMP. A second strategy consists in adding to the master problem only solutions that share at most τ sensors in common with the one with the best reduced cost found during PS solving process. As in the exact methods, the objective is to minimize the reduced cost of the solution to guarantee that only interesting columns are added to the problem. The GA uses an *elitist* strategy that keeps the best chromosomes at each generation, and uses them to improve the pool of solutions during the whole execution of GA. The characteristics of the GA are described below:

Chromosomes Each chromosome is defined as a vector $A_j = [a_1, a_2, a_3 \dots a_n]$ with $a_i \in \{0, 1\}$ where the value of 1 indicates that sensor i belongs to the subset and 0 otherwise.

Initial population An initial population of N_{pop} individuals is randomly generated and used to initialize the algorithm. For each individual in the population the initialization procedure indicates a random number $r \leq W$ of sensors to be active. Then, it randomly selects these r sensors to add them to the cover represented by the chromosome.

Fitness assignment For all the members in the population the fitness is calculated using the objective function of the PS, *i.e.*, the reduced cost criterion. Let $F_1 = [f(A_1), f(A_2), f(A_3), \dots, f(A_{N_{pop}})]$ be a vector containing the objective function of the solutions in the population and let F_{Max} and F_{min} be the maximum and minimum objective function values of the elements in the population. Then the fitness Fit_i could be defined as:

$$Fit_i = \frac{F_{Max} - f(A_i)}{F_{Max} - F_{min}}$$

Selection In order to generate a new population, two parent solutions are selected with a probability P_i calculated as:

$$P_i = \frac{Fit_i}{\sum_{i \in 1 \dots N_{pop}} Fit_i}$$

Crossover At each iteration, a new population is generated through crossover and mutation operators. For each couple of parents that were selected by using the selection operator, then they are crossed with a probability P_c to generate a single children that will mutate with a probability P_m . The crossover operator proposed uses the parent structure to define the activation state of each sensor in the child. If both parents have a sensor active, with a high probability P_{on} this sensor will be in the same state at the child. If just one parent has a sensor active, with a probability P_{med} this sensor will be active in the child. Finally, if none of the parents have a sensor active with a low probability P_{off} this sensor will be active on the child. The proposed GA then verifies that the new child is not already in the pool of elite solutions. If this is not the case, it is included and the process is repeated until the new population is complete. Otherwise, the recently generated child is discarded. In order to tune up the values for the probabilities inside the algorithm a large set of experiments was conducted in randomly generated instances of PS.

Mutation The mutation operator selects a random number of sensors to perform a bit-flip that changes the current state of these sensors.

Feasibility preservation For each individual generated, a feasibility check is performed after the crossover and mutation operations are performed. The columns that have more than W sensors active cannot enter to the new population. In the same way, duplicates are removed.

At each iteration of the CG, the GA is used during N_{iter} iterations. After this, the best solution is evaluated to check if it is profitable. If the reduced cost of this value is negative, the best κ columns with negative reduced cost are added to the RMP.

If there are no negative reduced cost individuals, the algorithm continues for another $N'_{iter} = 2N_{iter}$ iterations. If no reduced cost columns are found, the integer programming solver is used to find a new solution. If a new profitable solution is found, a new CG iteration is started. Otherwise, the optimal solution is obtained and CG stops.

3.3.2.4 Computational experiments

The methods are tested with the same instances as the stabilization procedures. For the GA a population of $N_{pop} = 50$ individuals is generated at each iteration and the best 10 individuals of each generation are stored. A testbed of instances of the PS was used to select the values for P_{on}, P_{med}, P_{off} , are selected as 0.95, 0.5 and 0.05 respectively. Similarly, the crossover probability is set at $P_c = 0.95$ and the mutation probability to $P_m = 0.03$. All the experiments are performed using the same set of parameters. In the intensification strategy, all the individuals of the population with negative reduced cost are returned to RMP; by contrast, in the diversification strategy only the interesting columns that share at most one sensor in common with the best reduced cost column found are returned.

Table 3.6 presents a comparison of the number of iterations obtained by using the intensification strategies. The columns labeled κ -Int and Div presents the results of the κ -Intensification and diversification approaches respectively. The experiments confirm that, by applying intensification strategies to CG, the number of iterations required to reach the optimal solutions is largely reduced. The results seems to show that intensification allows to reduce the number of iterations by tackling indirectly the degeneracy problem by, likely, providing more information to RMP. As a consequence, next iterations lead to the production of better quality columns which allows to improve the objective function more efficiently along CG process. Regarding to MCBB, previous results mean that CG might be improved by returning several solutions found in the solution process of PS, instead of one, contributing indirectly to deal with the problems that are the cause of the slow convergence.

Regarding the behavior of the dual variables, the use of intensification strategies produces an interesting behavior. In Figure 3.5 it is depicted the typical evolution of the dual variables values distance to the optimal one $\|\pi^j - \pi^*\|$. It was observed that, whereas in the first iterations the evolution of dual variables shows a behavior not too different to the one observed in CG without stabilization, a more stable behavior is observed after a few iterations have been carried out. The observed behavior seems to indicate that, by using intensification approaches and return more than a single column at each iteration a stable behavior of the dual variables is induced. Consequently, the use of stabilization strategies might be unnecessary.

As expected, the results show that the rough inclusion of a big group of columns to RMP improves the performance compared to the diversification strategy. This could be explained by the fact that, without using any other criterion to generate the columns neither deletion, the filtering of the columns is limiting the information included in

n	K	Rs	W	CG-CA				CG		
				κ -Int		Div		#Iters	Time (s)	
				#Iters	Time (s)	#Iters	Time (s)			
50	30	130	5	40.33	1.33	48.00	1.77	99.67	2.06	
			10	22.67	1.12	29.33	1.11	52.33	0.87	
			150	5	33.00	0.67	67.67	1.01	80.00	1.61
			10	5.67	0.81	38.33	0.65	39.33	0.77	
100	60	130	5	155.00	33.54	205.33	43.27	304.33	41.87	
			10	67.33	5.46	125.33	7.65	142.00	10.30	
			150	5	122.00	9.36	198.00	23.56	249.67	28.08
			10	36.00	2.79	65.67	12.02	99.67	7.51	
150	90	130	5	285.33	92.30	275.33	177.86	468.67	234.34	
			10	143.67	41.54	126.33	27.57	264.00	37.35	
			150	5	263.00	99.69	311.33	213.54	441.67	250.86
			10	64.00	29.66	93.67	21.66	134.67	23.73	
200	120	130	5	581.33	1995.65	48.67	2256.63	879.67	3518.33	
			10	232.00	71.78	158.67	77.65	365.00	102.29	
			150	5	420.67	1677.11	477.00	1257.45	653.00	2283.45
			10	102.00	48.93	140.33	60.67	181.33	61.19	
Average variation			-50.3%	-28.8%	-34.1%	-12.5%	-	-		

Table 3.6 – Comparison of number of iterations for κ -intensification and diversification techniques.

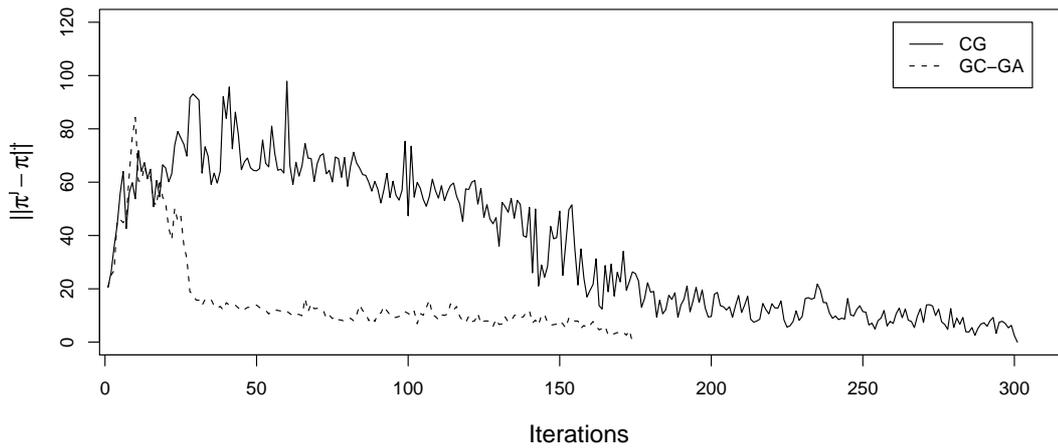


Figure 3.5 – Comparison of dual behavior in the CG without stabilization (CG) and the intensification strategy (CG-GA).

the RMP compared to the rough intensification approach. In this way, it is possible to conclude that the expansion of the RMP provides the CG with the ability to explore the solution space in more efficient ways than when it is limited to changes of one column each time.

3.3.3 Hybridizing stabilization and intensification strategies in column generation

Touati *et al.* [Moungla et al., 2010] briefly outline the relation between stabilization and diversification strategies for CG. As previously mentioned, the objective pursued by using stabilization methods is to try to compute an estimation of the optimal dual solution that will guide the solution process smoothly to the optimal primal solution in a reduced number of iterations. In other words, this method is intended to reduce the number of iterations by using an estimated approximation of dual function based on current best dual solution found. By contrast, by using diversification strategies the purpose is not to estimate this function, but to construct a good dual function global approximation based on the idea of providing a more rich source of information to the RMP.

As previously mentioned, the use of hybrid methods combining stabilization and other approaches often produces improvements in the convergence of CG [Benchimol et al., 2012]. The overall idea consists in taking advantage of the benefits offered by each one of the techniques by combining both into a single algorithm. In this study it is proposed to combine the κ -intensification strategy with a stabilization method to evaluate the effect over MCBB. The purpose of such hybridization is then to enrich the information available in RMP to approximate a good dual optimal solution that can be used to guide the solution in an efficient manner to the optimal solution for MCBB. We choose the most efficient DVS method reported in section 3.3.1.5, namely Du Merle' stabilization method. The proposal is to use a GA to generate several columns at each iteration of GA and return several of them to the stabilized version of RMP.

3.3.3.1 Computational experiments

As in the κ -Intensification approach, the hybrid strategy is tested in the same set of instances as the stabilization procedures. The initial dual box centers for DVS are selected as the dual variables values obtained through the initialization heuristic. The GA is used to solve PS using as input the dual variables values corresponding to the current optimal dual solution of the stabilized versions of RMP (3.33)-(3.41). Furthermore, the parameters of the GA are set to be the same previously used in the pure κ -Intensification strategy.

Table 3.7 compares the hybrid approach (S- κ -Int) with CG without stabilization and the pure κ -Intensification (κ -Int) strategy introduced in section 3.3.2.1. The methods are compared in terms of the number of iterations (#iters) and the computational time. The results show that, by using the combined approach, it is obtained an improvement compared to the κ -Intensification strategy and CG without stabilization. The proposed

strategy is shown to be able to reduce the number of iterations required to reach an optimal solution. Furthermore, the hybrid strategy reduces the average computational time required to solve MCBB.

n	K	Rs	W	CG-CA				CG		
				κ -Int		S- κ -Int		#Iters	Time (s)	
				#Iters	Time (s)	#Iters	Time (s)			
50	30	130	5	40.33	1.33	37.67	1.51	99.67	2.06	
			10	22.67	1.12	22.33	1.13	52.33	0.87	
			150	5	33.00	0.67	31.00	0.75	80.00	1.61
			10	5.67	0.81	7.00	0.41	39.33	0.77	
100	60	130	5	155.00	33.54	143.33	23.54	304.33	41.87	
			10	67.33	5.46	63.33	6.85	142.00	10.30	
			150	5	122.00	9.36	116.00	7.32	249.67	28.08
			10	36.00	2.79	42.67	3.29	99.67	7.51	
150	90	130	5	285.33	92.30	280.33	104.57	468.67	234.34	
			10	143.67	41.54	143.67	31.05	264.00	37.35	
			150	5	263.00	99.69	273.00	99.69	441.67	250.86
			10	64.00	29.66	63.33	24.00	134.67	23.73	
200	120	130	5	581.33	1995.65	578.00	2286.20	879.67	3518.33	
			10	232.00	71.78	217.00	74.56	365.00	102.29	
			150	5	420.67	1677.11	422.00	1465.11	653.00	2283.45
			10	102.00	48.93	37.67	50.32	181.33	61.19	
Average variation			-50.3%	-28.8%	-53.0%	-34.5%	-	-		

Table 3.7 – Comparison of number of iterations for κ -intensification and diversification techniques.

Although the method seems to perform better than the other intensification approaches, it is observed that the obtained reductions are close to the ones obtained by using a κ -Intensification approach. This could be partially explained by the fact that the use of intensification strategies significantly reduces the unstable behavior of the dual variables values in such a way that the use of DVS could be not required.

3.4 Conclusions and future work

This chapter evaluates the effect of stabilization and diversification strategies when applied in a CG scheme used to solve the MCBB. The obtained results allow to conclude that both strategies are able to accelerate the CG. The effects on the dual variables evolution is presented. It is possible to observe that, as expected, by using stabilization strategies the zig-zag effect observed in the dual variables is reduced compared to the CG without stabilization and so the number of iterations. Compared with CG without stabilization, intensification strategies do not present a different behavior of the dual variables values during the first iterations of CG. However, it is possible to conclude that intensification strategies naturally lead to a more stable behavior of the dual variables and, in general, help to improve convergence. Hence, it might indicate that the use of stabilization strategies became unnecessary when intensification is applied to MCBB.

The use of GA seems to be a simple and natural approach to solve the PS integrating κ -Intensification and diversification strategies. This seems to be confirmed with the numerical experiments. Regarding to MCBB, it was observed that, in general, in-

tensification strategies arise as the best strategy to improve the performance of CG. In addition, by using intensification without applying any filtering to the returned columns, *i.e.*, including diversification, the method seems to perform better. This result could be explained because more information is provided to RMP and it might reduce the undesirable effects of degeneracy.

A similar approach can be adopted to solve other problems in WSN, *e.g.*, the maximum network lifetime problem. It is possible to conclude that stabilization and intensification strategies are attractive methods to accelerate the convergence of CG approaches applied to WSN. Consequently, the techniques explored in the chapter can be easily extended when considering those problems.

Extensions to the MCBB in which connectivity constraints must be enforced or adjustable sensing ranges could be considered can be benefited of the use of these strategies. Intensification and acceleration strategies could report huge benefits when the time required to solve PS is high and it is desired to avoid performing useless iterations.

4

A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints

This chapter addresses the maximum network lifetime problem in wireless sensor networks with connectivity and coverage constraints. In this problem, the purpose is to schedule the activity of a set of wireless sensors, keeping them connected while network lifetime is maximized. Two cases are considered. First, the full coverage of the targets is required, and second only a fraction of the targets have to be covered at any instant of time. An exact approach based on column generation and boosted by GRASP and VNS is proposed to address both of these problems. Finally, a multiphase framework combining these two approaches is built by sequentially using these two heuristics at each iteration of the column generation algorithm. The results show that our proposals are able to tackle the problem efficiently and that combining the two heuristic approaches improves the results significantly.

4.1 Introduction

Wireless sensors are small devices with low energy consumption rates that are typically deployed to monitor some interesting phenomena, *e.g.*, surveillance, military applications, environmental monitoring, etc [Dargie and Poellabauer, 2010, Yick et al., 2008]. In wireless sensor networks (WSN) deployed to monitor targets, these devices work collaboratively or individually to collect information from the field and to deliver or spread the collected data to a remote base station through a multihop path of active sensors.

Energy consumption is a major concern for the implementation and deployment of WSN [Cardei et al., 2005, Slijepcevic and Potkonjak, 2002]. Furthermore, there exists an extended range of applications in which the replacement of sensors or the renewal of batteries is not feasible, like in hostile or contaminated environments. This fact stresses the necessity of designing efficiently schemes for the simultaneous use of sensors energy.

In order to keep the network operating as long as possible, a common strategy is to deploy more sensors than actually needed. Then, network lifetime can be extended by activating sequentially subsets of sensors able to meet the network requirements. The sensing range R_s is defined as the maximum distance a sensor can cover a target.

Two sensors are considered connected if the distance between them is less than the communication range R_c (in practice, $R_s \leq R_c$). Only the sensors from an active set are available for monitoring targets and transmitting the collected data. So, the optimal use of network energy can be obtained by identifying and creating schedules for the use of the sensors in the network.

In some applications, the complete collection of information originated in the targets is not a critical requirement. Thus, a threshold can be defined as the minimum level of coverage provided by the network, *i.e.*, the fraction α of targets that have to be covered at any instant of time. This characteristic provides the network with a bit of flexibility which, in addition, allows to increase its lifetime by neglecting some of the targets that are poorly covered and become a bottleneck limiting the network lifetime [Gentili and Raiconi, 2013].

In order to optimize the usage of the energy in WSN, researchers have addressed the maximum network lifetime problem (MLP) [Cardei et al., 2005, Deschinkel, 2011, Jia et al., 2008, Slijepcevic and Potkonjak, 2002]. This problem consists in maximizing the lifetime of a WSN whilst guaranteeing the coverage of a discrete set of targets. Specifically, a lot of effort has been devoted to solve the non-connected version of MLP. Thus, previous works provide a good starting point for the development of efficient approaches to solve new versions of MLP.

Recent researches show a growing interest in the use of exact approaches to solve optimization problems in WSN [Alfieri et al., 2007, Gu et al., 2009b, Rossi et al., 2012a]. Column generation (CG) has been largely used to address different versions of MLP. CG decomposes the problem in a restricted master problem (RMP) and a pricing subproblem (PS). The former maximizes lifetime using an incomplete set of columns, and the latter is used to identify new profitable columns. Gu *et al.* [Gu et al., 2011] have studied the coverage and scheduling problem in WSN. As maximum network lifetime problem with coverage constraints inherently involves time issues, the problem is represented by using a time-dependent structure that considers the coverage as a function of time and impose constraints on it. Then, they show that this kind of representation for MLP can always be converted into a pattern-based representation that points to maximize lifetime by using subsets of sensors (patterns) that satisfy the coverage requirement. As the number of feasible patterns grows exponentially with the number of sensors, the authors conclude that CG offers a natural way to address coverage and scheduling problems in WSN. Experimental results show that this approach is able to find optimal solutions to medium size instances of MLP. Moreover, recent researches show that this method can be improved by using heuristic approaches embedded in CG to solve the pricing subproblem [Rossi et al., 2012a, Singh et al., 2013].

When the connectivity constraint is also required, the problem is referred to as CMLP. In area coverage, a sufficient condition to guarantee connectivity is that the communication range R_c is at least twice the sensing range R_s ($R_c \geq 2R_s$) [Tian and Georganas, 2005]. However, Lu *et al.* [Lu et al., 2005] have shown that this property does not hold for target coverage, and have proposed a distributed heuristic to solve the problem. Further results presented by Singh *et al.* [Singh et al., 2013] show that, for the Q-coverage version of CMLP, in which each target has to be covered by at

least Q active sensors at any time, an efficient approach can be generated by relaxing connectivity constraints in the PS. In other words, by solving the problem as in MLP and trying to restore connectivity if necessary.

CMLP has been addressed by Cardei and Cardei [Cardei and Cardei, 2008a] who propose three different heuristic approaches. First, an integer programming model of the problem which is solved to create, through a heuristic, an energy-efficient scheme of the problem. Then, the authors propose two greedy heuristics to create iteratively set covers in centralized and distributed manners respectively. Raiconi and Gentili [Raiconi and Gentili, 2011] propose a greedy procedure namely CMLP-Greedy and two variants embedded in a greedy randomized adaptive search procedure (GRASP) to find pattern based solutions seeking to maximize network lifetime. Furthermore, the authors compare their results with an exact decomposition based approach using CG and show that their method is computationally efficient and, in addition, is able to find near optimal solutions in most cases.

Zhao and Gurusami [Zhao and Gurusamy, 2008] propose to solve CMLP by modeling the problem as a maximum cover tree problem (MCTP). In their proposal, the idea is to find a collection of subtrees and timings to maximize network lifetime. The authors show that MCTP is NP-Complete by reduction of 3-SAT problem. The authors propose an upper bound to the network lifetime and propose two heuristics to solve the problem.

Several mixed approaches combining heuristic and exact approaches have been introduced recently to solve optimization problems in various of contexts [Blum et al., 2011, Maniezzo et al., 2010]. Heuristic approaches combining CG or Lagrangian relaxation with (meta)heuristic approaches are shown to be successful in a lot of applications. Recently, Rossi *et al.* [Rossi et al., 2012a] presented an efficient implementation of a genetic algorithm based CG to extend lifetime and maximize coverage in wireless sensor networks under bandwidth constraints. The authors show that the use of metaheuristic methods to solve PS in the context of CG allows to obtain optimal solutions quite fast, and to produce high quality solutions when the algorithm is stopped before returning an optimal solution.

In this chapter an exact multi level approach based on CG is proposed to solve the CMLP. Our proposal is to speed up the solution process by embedding two heuristic approaches within the CG framework. First, a greedy randomized adaptive search procedure (GRASP) [Feo and Resende, 1995] is proposed to solve PS. This approach relaxes connectivity constraints, so a repair procedure is necessary. Then, when GRASP approach fails to find a profitable solution to PS, a variable neighborhood search (VNS) heuristic [Hansen and Mladenović, 2001] is attempted for finding profitable columns. Finally, if both heuristics are unable to find a profitable solution, integer linear programming (ILP) is used to solve PS. It is also used for proving optimality of the current RMP solution at the very end of the search. An extension of the problem, namely α -CMLP, is also considered. It consists in replacing the full coverage requirement by a constraint for enforcing a minimum quality of service. Thus, it is possible to neglect a fraction $1 - \alpha$ of the targets, which allows to extend lifetime.

This chapter is organized as follows. Section 4.2 introduces the problem description and the decomposition approach used to solve α -CMLP. A detailed description of the

proposed approach is presented in Section 4.3. The results obtained through the use of the proposed methods and a detailed analysis of the computational experiments are reported in Section 4.4. Finally, conclusions and future work are presented in Section 4.5.

4.2 The maximum network lifetime problem under coverage and connectivity constraints

Consider a set $\mathcal{K} = \{k_1, \dots, k_m\}$ of targets with known locations and a set $\mathcal{S} = \{s_1, \dots, s_n\}$ of sensors deployed to cover the targets. If the distance between a sensor node and a target is less than its sensing range R_s , then this sensor is able to cover the target and an *observation link* exists. The sensor nodes collect and (re)transmit the information to other sensor nodes within their communication range R_c (*communication link*). All the information generated by the targets must be collected by a single sink node r . A sensor is able to send the information to the sink node only if a communication link exists between them, otherwise the information have to be addressed indirectly through a multi hop path of sensors.

Let E be the set of all pairs $e(u, v)$ such that a communication link exists between the elements $u, v \in \mathcal{S} \cup \{r\}$ or an observation link exists between the elements $u \in \mathcal{K}$ and $v \in \mathcal{S}$. A feasible cover $C_j \subseteq \mathcal{S}$ is a subset of sensors such that for at least $\lceil \alpha |\mathcal{K}| \rceil$ targets, there exists a communication link $e(u, v)$ between $u \in \mathcal{K}, v \in C_j$ and there exists a path between the elements of C_j and r . The set of all the feasible covers of \mathcal{S} is denoted by $\Omega = \{C_1, C_2, \dots, C_\ell\}$.

Variable t_j is the time during which cover C_j is used. The α -connected maximum network lifetime problem (α -CMLP) is defined as finding a collection of pairs (C_j, t_j) , such that network lifetime, $\sum_{j \in \Omega' \subseteq \Omega} t_j$, is maximized without exceeding the battery capacity b_{s_i} of the sensors s_i .

Let $y_{s_i j}$ be a binary parameter that is set to 1 iff sensor s_i is active in cover C_j . α -CMLP can be formulated as the following linear program:

$$\text{Maximize: } \sum_{C_j \in \Omega} t_j \quad (4.1)$$

Subject to:

$$\sum_{C_j \in \Omega} y_{s_i j} t_j \leq b_{s_i} \quad \forall s_i \in \mathcal{S} \quad (4.2)$$

$$t_j \geq 0 \quad \forall C_j \in \Omega \quad (4.3)$$

The objective (4.1) is to maximize the network lifetime by using a collection of covers C_j that meet the connectivity and coverage constraints. The set of constraints (4.2) is used to guarantee that battery constraints of the sensors are respected. Constraints (4.3) are the non-negativity constraints.

4.2.1 Decomposition approach

The model (4.1-4.3) is linear and is known to be easy to solve with a linear programming solver [Deschinkel, 2011, Gentili and Raiconi, 2013]. By contrast, the enumeration of all the feasible covers C_j is generally impossible, as the number of such covers grows exponentially with the number of sensors $\mathcal{O}(2^{|\mathcal{S}|})$ which stresses the need for intelligent strategies to identify profitable covers. This can be achieved by using the reduced cost of the decision variables of t_j for all $j \in \Omega$.

In order to solve α -CMLP, our proposal is to divide the problem into two. First, a *restricted master problem* (RMP), containing a reduced set of the feasible columns $\Omega' \subseteq \Omega$, is used to identify the timings for the covers in Ω' . In a second stage, a *pricing subproblem* (PS) is used to identify new profitable covers in Ω , that will be added to Ω' . In order to identify profitable columns, the reduced cost criterion is used iteratively to generate and add columns to RMP. For each new cover the reduced cost is evaluated. If it is strictly positive, which means that the cover is interesting, it is added to RMP and a new CG iteration is performed. The CG process stops when no more profitable columns are found.

4.2.1.1 Pricing subproblem

The pricing subproblem is to identify connected structures using different subsets of sensors so that the network lifetime can be extended. Then, PS purpose is to find a minimum cost tree spanning a fraction α of the targets and the base station, *i.e.*, guaranteeing connectivity.

In order to find profitable covers, we propose an extension of the flow model presented by Raiconi and Gentili [Raiconi and Gentili, 2011] in which partial coverage is considered. The authors propose to formulate the problem as a single-commodity flow to find a tree connecting all the active sensors. In their model, the authors simulate a flow leaving the base station which value is equal to the number of active sensors that consume this flow. Let \check{y}_u be a binary variable that is set to 1 if sensor s_u is part of cover C_j being generated at iteration j of CG. Let z_m be a binary variable that is set to one if and only if target $k \in \mathcal{K}$ is covered by a sensor in C_j . The binary variables x_{uv} are used to decide if a communication link is established between sensors s_u and s_v . In the same way, the integer variables f_{uv} are used to identify the flow passing through the communication link $e(u, v)$. For each sensor s_v , the dual variable associated with constraint (4.2) is denoted by π_v .

By using the above notation, the PS can be modeled as follows:

$$\text{Maximize: } 1 - \sum_{s_u \in \mathcal{S}} \check{y}_u \pi_u \quad (4.4)$$

Subject to:

$$\sum_{s_u \in \mathcal{S} | \exists e(r,u)} f_{ru} = \sum_{s_u \in \mathcal{S}} \check{y}_u \quad (4.5)$$

$$\sum_{s_v \in \mathcal{S} | \exists e(v,u)} f_{vu} - \sum_{s_v \in \mathcal{S} | \exists e(u,v)} f_{uv} = \check{y}_u \quad \forall s_u \in \mathcal{S} \quad (4.6)$$

$$\sum_{s_u \in \mathcal{S} | \exists e(u,v)} x_{uv} = \check{y}_v \quad \forall s_v \in \mathcal{S} \quad (4.7)$$

$$x_{uv} \leq f_{uv} \leq x_{uv} |\mathcal{S}| \quad \forall s_u, s_v \in \mathcal{S} \quad (4.8)$$

$$\sum_{s_u \in \mathcal{S} | \exists e(u,k)} \check{y}_u \geq z_k \quad \forall k \in \mathcal{K} \quad (4.9)$$

$$\sum_{k \in \mathcal{K}} z_k \geq \lceil \alpha |\mathcal{K}| \rceil \quad (4.10)$$

$$f_{uv} \in \mathbb{Z}^+ \cup \{0\} \quad \forall s_u, s_v \in \mathcal{S} \quad (4.11)$$

$$\check{y}_u \in \{0, 1\} \quad \forall s_u \in \mathcal{S} \quad (4.12)$$

$$x_{uv} \in \{0, 1\} \quad \forall s_u, s_v \in \mathcal{S} \quad (4.13)$$

$$z_k \in \{0, 1\} \quad \forall k \in \mathcal{K} \quad (4.14)$$

As previously mentioned, the purpose of PS is to identify profitable network structures based on the reduced cost criterion (4.4). Equation (4.5) ensures that the flow offered by the base station is equal to the number of active sensors consuming the information. Flow balance constraints are imposed in (4.6). Furthermore, if a sensor is part of C_j , then exactly one entering communication link has to be active (4.7). Bounds are imposed on the flows by using the set of constraints (4.8). Constraints (4.9-4.10) are used to guarantee a minimum level of target coverage.

4.3 Solving the pricing subproblem

Integer programming has been used to solve the full coverage version of PS [Raiconi and Gentili, 2011]. Nonetheless, experimental results show that exact methods become inefficient even for small problem sizes. Furthermore, by following a CG framework the number of required iterations is expected to grow with the problem size as the number of feasible covers also grows.

Angelopoulos [Angelopoulos, 2006] and Li *et al.* [Li et al., 2009] model the PS as a node weighted Steiner tree and show that it is NP-Hard. Although polynomial algorithms exist in the literature to approximate this problem and some of its variants, these results are not good enough for being used in a CG algorithm, because attractive columns might be missed, leading to a premature convergence of the CG approach.

In order to speed up the CG approach and obtain optimal solutions for α -CMLP, we propose to solve PS by using a multiphase approach. An overview of the proposed method is presented in Figure 4.1. At each iteration of CG, the method attempts to find

interesting solutions for PS by resorting to three methods. First, a GRASP heuristic is proposed to solve PS. This method is based on the simple idea of addressing PS by relaxing connectivity constraints, and then repairing the solution. If a profitable column is found, it is returned to RMP and a new iteration of CG is performed. Several columns can be returned at each iteration of CG as an strategy to accelerate the convergence of the technique [Lubbecke and Desrosiers, 2005, Mounpla et al., 2010]. If GRASP fails to find a profitable cover, a VNS approach is executed for solving PS without relaxing connectivity and coverage constraints. If this also fails, then ILP is used to solve PS. If no profitable column exists, the current solution to RMP is proved optimal for α -CMLP.

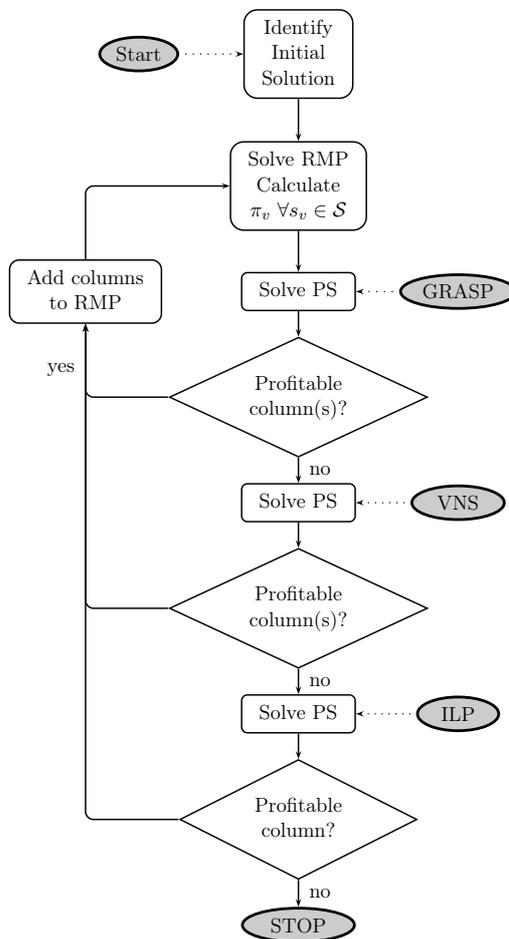


Figure 4.1 – Multiphase column generation approach.

Pure heuristic approaches can be obtained as well by not solving the ILP formulation of PS, and by applying sequentially both heuristics as mentioned above (or even by running a single heuristic). These variants are considered in this chapter. A detailed description of the heuristic approaches is presented below.

4.3.1 A GRASP approach to solve the pricing subproblem

As previously mentioned, the first strategy consists in solving a relaxed version of PS where the connectivity constraint is ignored. The problem can then be seen as a weighted set covering problem where the cost of using a sensor is equal to its dual value. The purpose is to find the minimum cost subset of sensors which is able to meet the coverage constraints. Then, in order to produce feasible covers, a repair procedure must be performed for non-connected solutions.

In this chapter, a GRASP algorithm is used to solve the relaxed version of PS. An overview of the proposed GRASP heuristic is presented in Algorithm 4.1. Note that the objective of the algorithms is to minimize $\sum_{u \in \mathcal{S}} \check{y}_u \pi_u$ for maximizing the objective value of PS (4.4). The GRASP algorithm uses a greedy randomized constructive phase (line 4) to compute an initial solution \hat{C}_j which is improved through a local search procedure (line 5). The constructive phase is based on the algorithm proposed by Chvátal [Chvátal, 1979]. A feasibility check is performed to evaluate connectivity of the solutions (line 7). If the solution is connected and its cost is lower than the best solution found so far, the solution is stored. Otherwise a repair operator is performed (line 10). The reparation consists in computing the main connected component (i.e., the one that contains the base station), and the other connected components. Connectivity is then enforced by adding the sensors on the shortest path to the main connected component. The repair completes when the desired level of coverage is reached through the recently included sensors, or when the solution is connected.

Algorithm 4.1: Set Covering GRASP

```

1 GRASP_SC( $\pi_v, \alpha$ )
2  $C_j \leftarrow \emptyset, \hat{C}_j \leftarrow \emptyset, time \leftarrow 0, iter \leftarrow 0, f(C_j) \leftarrow \infty$ 
3 while  $time \leq Max\_time$  &  $iter \leq MaxIters$  do
4    $\hat{C}_j \leftarrow Greedy\_Construction(\pi_v, \alpha, \beta)$ 
5    $\hat{C}_j \leftarrow Sol\_Improvement(\hat{C}_j)$ 
6   if  $f(\hat{C}_j) \leq f(C_j)$  then
7     if  $Feasibility\_Check(C_j)$  then
8        $C_j \leftarrow \hat{C}_j, iter \leftarrow 0$ 
9     else
10       $Shortest\_Path\_Repair(C_j)$ 
11      if  $f(\hat{C}_j) \leq f(C_j)$  then
12         $C_j \leftarrow \hat{C}_j, iter \leftarrow 0$ 
13      end
14    end
15  end
16   $time \leftarrow Update\_time(), iter \leftarrow iter + 1$ 
17 end
18 return  $C_j$ 

```

4.3.1.1 Greedy randomized construction

The constructive phase of the algorithm is based on the greedy heuristic for the weighted set covering problem proposed by Chvátal [Chvátal, 1979], it is described in Algorithm 4.2. Let T_v denotes the set of targets covered by sensor v , the method iteratively selects the next sensor to be included in a cover based on the ratio ind_v between the reduced cost of sensor $s_v \in \mathcal{S}$ and the number of uncovered targets that it covers (lines 4-5). Then, it creates a restricted candidate list RCL containing only the elements within a fraction β of lowest index ignoring those sensors not covering any uncovered target (lines 7-9). A random selection is performed among the elements in RCL (line 10) and the selected sensor is included in the current cover. The process stops when the required coverage level is reached (line 3).

Algorithm 4.2: Greedy Randomized Construction

```

1 Greedy_Randomized_Construction( $\pi_v, \alpha, \beta$ )
2  $C_j \leftarrow \emptyset, \mathcal{K}_{cov} \leftarrow \emptyset, \mathcal{K}_{unc} \leftarrow \mathcal{K}, \mathcal{S}_{av} \leftarrow \mathcal{S}$ 
3 while  $|\mathcal{K}_{cov}| < \alpha|\mathcal{K}|$  do
4   for  $s_v \in \mathcal{S}_{av}$  do
5      $ind_v = \frac{\pi_v}{|T_v \cap \mathcal{K}_{unc}|}$ 
6   end
7    $c_{min} = \min_{s_v \in \mathcal{S}_{av}} ind_v$ 
8    $c_{max} = \max_{s_v \in \mathcal{S}_{av}} ind_v$ 
9    $RCL \leftarrow \{s_v \in \mathcal{S}_{av} | \pi_v \leq c_{min} + \alpha(c_{max} - c_{min})\}$ 
10   $sel \leftarrow random\_selection(RCL)$ 
11   $C_j \leftarrow C_j \cup sel$ 
12   $\mathcal{S}_{av} \leftarrow \mathcal{S}_{av} \setminus sel$ 
13  for  $u \in \mathcal{K}_{unc}$  do
14    if  $\exists e(sel, u)$  then
15       $\mathcal{K}_{unc} \leftarrow \mathcal{K}_{unc} \setminus u$ 
16    end
17  end
18  for  $v \in \mathcal{S}_{av}$  do
19    if  $T_v \cap \mathcal{K}_{unc} = \emptyset$  then
20       $\mathcal{S}_{av} \leftarrow \mathcal{S}_{av} \setminus v$ 
21    end
22  end
23 end
24 return  $C_j$ 

```

4.3.1.2 Solution improvement

A local search is performed for improving the solution by trying to remove the sensors that do not contribute to the solution feasibility. A best improvement strategy is performed by evaluating first the removal of the sensors with the highest π_v value.

4.3.2 A VNS approach to solve the pricing subproblem

Variable neighborhood search is a well known metaheuristic introduced by Hansen and Mladenović [Hansen and Mladenović, 2001] consisting on the systematic exploration of several neighborhoods for solving optimization problems. The method exploits several ideas that allow to address problems in the context of global optimization. First, the method is able to obtain solutions that are locally optimal under several neighborhood structures. In addition, the method provides strategies to escape from locally optimal solutions so it enables the evaluation of unexplored regions of the feasible space.

In order to efficiently solve PS, we propose to apply a basic variable neighborhood search heuristic (BVNS). This variant of VNS relies on a combination of stochastic and deterministic changes of neighborhood to explore the search space. An overview of BVNS is presented in Algorithm 4.3.

Let $\mathcal{N}_k(C_j)$ denote the k^{th} neighborhood of a feasible solution C_j for PS. A shake function is used to select a random solution $C'_j \in \mathcal{N}_k(C_j)$. Once the new point has been selected, a local search procedure is executed until no better solution is found. Then, a change of neighborhood is performed if the process falls into a local optimum and the neighborhood is unable to provide means to escape from it, otherwise, the process starts again with the first neighborhood. This process is repeated until a maximum number of iterations with no improvement is performed or the running time reaches the time limit.

Algorithm 4.3: Basic variable neighborhood search

```

1 BVNS(TimeLimit, MaxIters,  $\alpha$ ,  $\pi_v$ )
2  $S_{av} \leftarrow \emptyset, S_{act} \leftarrow \mathcal{S}, \mathcal{K}_{cov} \leftarrow \emptyset, s_0 \leftarrow r$ 
3  $C_j \leftarrow \text{Greedy\_DFS}(S_{av}, S_{act}, \mathcal{K}_{cov}, \pi_v, G, s_0, \alpha)$ 
4  $k \leftarrow 1$ 
5 while  $time < TimeLimit$  &  $iter \leq MaxIters$  do
6    $\hat{C}_j \leftarrow \text{shake}(C_j, k)$ 
7    $\check{C}_j \leftarrow \text{Local\_Search}(\hat{C}_j, k)$ 
8   if  $f(\check{C}_j) < f(C_j)$  then
9      $C_j \leftarrow \check{C}_j$ 
10     $k \leftarrow 1$ 
11     $time \leftarrow \text{Update\_Time}()$ 
12  else
13     $k \leftarrow k + 1$ 
14     $time \leftarrow \text{Update\_Time}()$ 
15  end
16 end
17 return  $\vec{C}_j$ 

```

4.3.2.1 Initial solution

In general, any connected set able to cover the whole set of targets and to send the information to the base station is useful. However, a greedy heuristic is intended to provide the local search procedures with a good starting point to seek for a high quality solution.

The proposed approach is a recursive algorithm used to add new sensors to an initial tree. A general description of the algorithm is outlined in Algorithm 4.4. The algorithm starts with an initial tree consisting only of the base station node. Next, unconnected sensors are added to the tree following a deep first search (DFS) strategy that prefers the sensors with the lowest dual variable value π_v associated to the battery limit constraints (4.2) and related to the set of sensors (lines 3-4). Each time that a new sensor is added to the tree, the targets sharing an observation link with it are marked as covered (lines 6-7). The procedure completes when a fraction α of the targets is covered.

Algorithm 4.4: Greedy depth first search (recursive)

```

1 Greedy_DFS( $S_{av}, S_{act}, \mathcal{K}_{cov}, \pi_v, G, s_0, \alpha$ )
2 while  $|\mathcal{K}_{cov}| < \alpha|\mathcal{K}|$  do
3    $s_0 \leftarrow \underset{v}{\operatorname{argmin}}\{\pi_v | v \in S_{av} \wedge \exists e(S_0, v)\}$ 
4    $S_{act} \leftarrow S_{act} \cup s_0$ 
5    $S_{av} \leftarrow S_{av} \setminus s_0$ 
6   for  $u \in \mathcal{K} | \exists e(u, s_0)$  do
7      $\mathcal{K}_{cov} = \mathcal{K}_{cov} \cup u$ 
8   end
9   GR_DFS( $S_{av}, S_{act}, \mathcal{K}_{cov}, \pi_v, G, s_0, \alpha$ )
10 end
11 return  $S_{act}$ 

```

4.3.2.2 Local search

The proposed local search consists in selecting an initial solution \hat{C}_j and an improvement direction, by performing moves leading to better solutions belonging to the neighborhood $\mathcal{N}_k(\hat{C}_j)$. Two types of strategies can be considered, *best improvement* and *first improvement*. The former evaluates the whole set of solution members of $\mathcal{N}_k(\hat{C}_j)$ and selects the best one as the next starting point. The latter performs a move each time that an improvement direction is detected. In this chapter, a first improvement strategy is proposed to explore the selected neighborhoods.

Algorithm 4.5: First improvement local search

```

1 Local_Search( $\hat{C}_j, k$ )
2 repeat
3    $\check{C}_j \leftarrow \hat{C}_j, i \leftarrow 0$ 
4   repeat
5      $i \leftarrow i + 1$ 
6     if  $f(\hat{C}_j^i) < f(\hat{C}_j) \wedge \hat{C}_j^i \in \mathcal{N}_k(\hat{C}_j)$  then
7        $\hat{C}_j \leftarrow \hat{C}_j^i$ 
8     end
9   until  $f(\hat{C}_j) < f(\check{C}_j)$  or  $i = |\mathcal{N}_k(\check{C}_j)|$ 
10 until  $\check{C}_j = \hat{C}_j$ 
11 return  $\hat{C}_j$ 

```

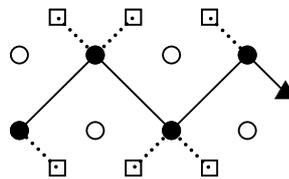
A fast exploration of the neighborhoods is obtained by avoiding checking the feasibility on the neighbors that are not profitable. This is possible by considering as interesting only those moves that are able to reduce the objective function and satisfy the quality of service and connectivity constraints of the network. We can identify the profitable neighbors before the evaluation of feasibility by using a simple mathematical relation. Let $\mathcal{S}' \subseteq \mathcal{S}$ be a subset of sensors for which the activation status is modified in a feasible solution $\hat{C}_j = \{y_{1j}, y_{2j}, y_{3j}, \dots, y_{|S|j}\}$, where the elements y_v are binary and take the value 1 if sensor s_v belongs to cover C_j . Let ΔC_j be the variation of costs in that solution after modifying the activation status of the sensors in \mathcal{S}' . Since $\Delta C_j = \sum_{v \in \mathcal{S}'} (1 - 2y_v)\pi_v$, only those modifications of a current solution producing a negative ΔC_j value are considered interesting. Then, a feasibility check can be performed to evaluate if the produced exchange maintains the required level of coverage.

4.3.2.3 Neighborhoods

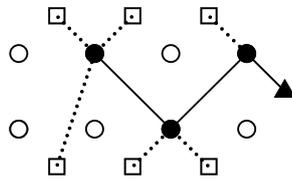
Using an initial solution as starting point, the local search procedures are required to evaluate moves and modifications of the current solution that allow to improve the objective function, *i.e.*, the network reduced cost. Consider $\hat{C}_j^1 = \{y_{1j}^1, y_{2j}^1, y_{3j}^1, \dots, y_{|S|j}^1\}$ and $\hat{C}_j^2 = \{y_{1j}^2, y_{2j}^2, y_{3j}^2, \dots, y_{|S|j}^2\}$ representing two valid solutions able to guarantee the coverage and the connectivity constraints in a WSN. The Hamming distance [Hamming, 1950] between the two solutions \hat{C}_j^1 and \hat{C}_j^2 is defined by $d(\hat{C}_j^1, \hat{C}_j^2) = \sum_{v \in \mathcal{S}} (y_v^1 \oplus y_v^2)$. Then, a **k-neighbor** solution of \hat{C}_j^1 is a feasible solution \hat{C}_j^2 such that $d(\hat{C}_j^1, \hat{C}_j^2) = k$. In other words, the number of sensors in which two sets differ is equal to k .

In this chapter we consider two neighborhoods for which $k \in \{1, 2\}$. First, A *remove neighborhood* ($k = 1$) is proposed to check for useless sensors which are not required to meet the coverage and connectivity constraints (see Figure 4.2b). Since all the associated dual variables are greater than or equal to zero, a best improvement strategy can easily be applied by considering the elements to remove according to the decreasing dual variable value associated with them. A *remove-insert neighborhood* ($k = 2$) is also considered. In this case one active sensor is replaced by an inactive sensor with a

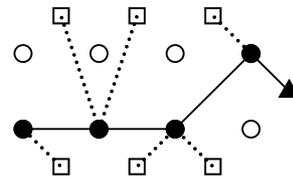
lower cost if it is able to keep the connectivity and coverage requirements (see Figure 4.2c). In general, similar neighborhood structures can be applied to explore a larger or structurally different portion of the solution space. For example, in order to make a fast exploration it could be useful to sample the solution space around the current best solution by randomly selecting a subset $\mathcal{S}' \subseteq \mathcal{S}$ of sensors such that $|\mathcal{S}'| = k$ and by modifying their activation status. Finally, the connectivity could be evaluated by performing a DFS between the active subsets of sensors.



(a) Initial solution.



(b) Remove neighborhood.



(c) Remove-Insert neighborhood.

Figure 4.2 – Neighborhood structures considered in the VNS approach for PS.

4.3.2.4 Shake

In order to provide a mechanism to allow the VNS procedure to escape from local optimal solutions and diversify the search, a shake function is proposed. This function takes as input an initial set of sensors that satisfy the coverage and connectivity requirements and modifies it to obtain a new solution. Several variations of this method can be considered. In this chapter, we propose the randomization of the VNS algorithm by including in a feasible subset a few more sensors that were not active before. Then, a new local search process starts from this random point using the same neighborhoods as previously mentioned.

4.4 Computational experiments

The proposed approaches are implemented in C++ and executed on an Intel Core i-5 processor at 1.6 GHz with 2 GB of RAM running under OS-X Lion. The Gurobi optimization engine is used to solve RMP and PS. Two main groups of instances were considered. First, the approaches are executed on the instance set proposed by Raiconi and Gentili [Raiconi and Gentili, 2011] for the full coverage case. Additionally, a new set of instances is considered to evaluate the performance of the method on different scenarios varying the ratio between the sensing and communication ranges. At each iteration of CG, GRASP or VNS are performed for a maximum duration of 1 second. Additionally, the algorithms are stopped if $|S|/2$ iterations are performed without improvement. The column generation process starts by finding an initial set of covers used to initialize the master problem. In a standard manner, a trivial solution consisting in only one column containing the whole set of sensors is used.

The use of stabilization strategies [Amor et al., 2009, Du Merle et al., 1999, Marsten et al., 1975] was considered to accelerate the convergence of the CG. However, the use of these strategies was discouraged by the fact that this problem does not exhibit unstable behavior for the dual variables numerical value from an iteration to the next one, neither a remarkable *tailing-off* effect, *i.e.*, while a near-optimal solution is approached considerably fast, the improvement of the objective function is slow in the last iterations. Figure 4.3 presents the typical evolution of the objective function (network lifetime) along successive iterations of CG. It can be observed that the network lifetime is rarely stalling, and keeps increasing by a non marginal amount even at the end of the search.

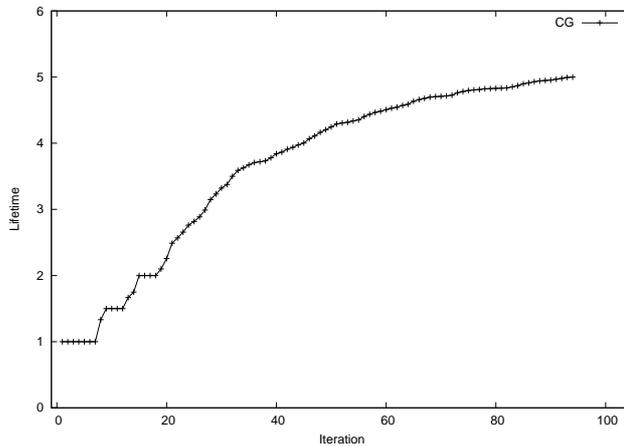


Figure 4.3 – Evolution of the objective function for CMLP.

Some experiments demonstrated that degeneracy problems can appear in some instances when the sensing and communication ranges are enlarged. Then, in order to evaluate the performance of stabilization and intensification approaches, the connectivity constraint was neglected. In stabilization the purpose is to control the behavior of the dual variables values (π_v), associated to convergence issues in CG, by limiting the

set of values that they can take at each iteration of CG. As RMP initially contains only partially the information relevant to the columns useful to extend network lifetime, the purpose of intensification is to provide more (probably diverse) information to RMP that help the CG process identifying better columns and reducing the number of iterations. This research considered the implementation of the BoxStep method [Marsten et al., 1975], and the generalization proposed by Du Merle *et al* [Du Merle et al., 1999]. The selection of the parameters for these methods was based on the specialized literature [Tran et al., 2006]. Moreover, we considered an intensification strategy in which at most 50 profitable columns are returned to the master problem at each iteration. The selection is based on the reduced cost criterion (Eq. 4.4).

Figure 4.4 presents the evolution of the euclidean distance to the optimal dual solution $\|\pi^* - \pi_j\|$ along successive iterations of CG for several intensification and stabilization strategies. As it could be observed, the use of stabilization strategies was useful to reduce the number of iterations required to reach an optimal solution for the problem. In the same way, intensification (and diversification) strategies offer a substantial reduction in the number of iterations required to reach the optimal solution for the problem. Moreover, our findings show that it presents as well a stabilizing effect reducing the large oscillations observed in the behavior of the dual variables in CG. The value at each iteration of the π_v variables are expected to be in the range $[0, 1]$. Although the knowledge of these values can be exploited by stabilization approaches, the results show that the use of intensification strategies can be used to accelerate the convergence. As a consequence, along this chapter the application of the described intensification strategy is adopted to accelerate the convergence of the proposed approach.

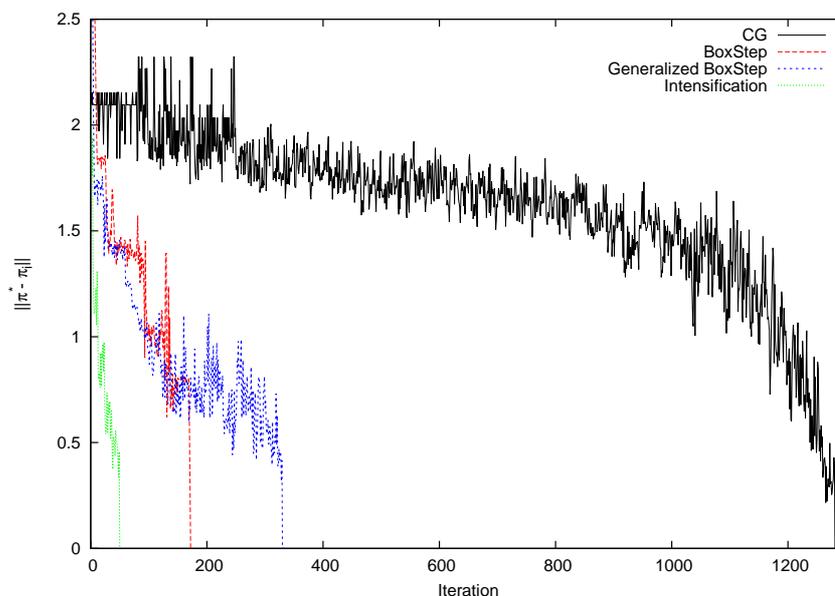


Figure 4.4 – Dual variables values behavior for several stabilization and intensification strategies.

Table 4.1 presents a comparison between the approaches introduced in this chapter and the heuristic approaches proposed by Raiconi and Gentili to address α -CMLP when $\alpha = 1$. Columns *LT* and *Time* report respectively the best lifetime found and the computational time for each approach. Labels CMLP-Greedy and CMLP-GRASP refer to the heuristics proposed by Raiconi and Gentili [Raiconi and Gentili, 2011]. Similarly, CG-EXACT refers to the results obtained through the multiphase exact approach proposed in this chapter and CG-MULTI reports the results obtained when it is applied as a pure heuristic approach (*i.e.*, ILP is never used). Finally, CG-CNS and CG-GRASP refers to the implementation of the CG using the VNS and GRASP heuristics, respectively, to generate the columns introduced in Section 4.3.1. In order to have a fair comparative study, the CPU times reported by Raiconi and Gentili have been scaled according to the Linpack benchmarks [Longbottom, 2012]. Bold font is used to highlight the fastest approaches among those that return an optimal solution. Of course, only CG-EXACT is able to prove optimality.

S	K	CMLP-Greedy		CMLP-GRASP		CG-EXACT		CG-MULTI		CG-VNS		CG-GRASP	
		LT	Time (s)	LT	Time (s)	LT	Time (s)	LT	Time (s)	LT	Time (s)	LT	Time (s)
50	15	2.0	0.02	2.00	3.01	2.00	0.51	2.00	0.04	2.00	0.06	2.00	0.24
		2.0	0.02	2.50	3.71	2.50	0.73	2.50	0.19	2.44	0.12	2.26	0.02
		3.0	0.02	3.00	3.33	3.00	0.45	3.00	0.05	3	0.02	3.00	0.03
		4.0	0.03	4.00	6.18	4.00	0.42	4.00	0.03	4	0.032	4.00	0.02
		3.0	0.02	4.00	7.70	4.66	0.71	4.66	0.37	4.66	0.84	4.58	0.34
75	15	5.0	0.05	6.00	24.37	7.00	1.61	7.00	0.64	7	2.43	6.99	0.60
		4.0	0.06	4.00	12.24	4.00	0.88	4.00	0.07	4	0.30	4.00	0.04
		3.0	0.03	3.00	4.10	3.00	1.15	3.00	0.51	3	0.03	3.00	0.03
		6.0	0.08	7.00	26.55	7.00	1.27	7.00	0.70	7	0.34	6.68	0.66
		6.0	0.04	6.00	19.31	7.00	1.37	7.00	1.01	7	1.36	6.80	0.58
100	15	8.0	0.11	9.00	33.20	10.00	6.31	10.00	2.59	10	11.22	10.00	2.54
		6.0	0.23	7.00	33.11	7.00	2.13	7.00	0.27	7	1.61	7.00	0.24
		6.0	0.06	7.00	17.48	7.00	1.72	7.00	0.12	7	0.24	7.00	0.09
		9.0	0.17	9.00	46.21	9.00	2.42	9.00	0.31	9	1.86	9.00	0.27
		8.0	0.20	8.00	86.49	8.00	3.45	8.00	0.26	8	1.07	8.00	0.19
150	15	12.0	0.25	15.00	69.22	17.00	87.84	17.00	54.83	17	99.27	15.50	5.37
		11.0	0.33	13.00	67.32	14.00	15.12	14.00	8.38	14.00	25.57	13.46	2.24
		10.0	0.17	12.00	30.37	16.00	14.45	16.00	5.77	16.00	25.48	16.00	5.69
		12.0	0.63	13.00	125.56	13.00	5.70	13.00	0.57	13	6.61	13.00	0.51
		13.0	0.29	14.00	131.02	14.00	5.84	14.00	0.76	14	3.13	14.00	0.62
200	15	18.0	0.58	20.00	160.51	25.00	95.04	25.00	78.00	24.79	97.13	16.21	1.78
		17.0	0.85	18.00	141.21	20.00	11.54	20.00	10.97	20	16.61	19.46	5.74
		13.0	0.27	14.95	156.00	19.00	17.12	19.00	15.24	19	42.18	18.73	6.38
		16.0	1.30	18.00	225.03	18.00	32.68	18.00	2.08	18	24.84	18.00	1.89
		18.0	0.70	19.00	218.34	19.00	7.00	19.00	2.96	19	9.513	19.00	2.93
#Opt		7		15		25		25		23		15	
Av. Time		0.26		66.06		12.70		7.47		14.87		1.56	
Av. Deviation		14.2%		5.9%		0.0%		0.0%		0.06%		2.8%	

Table 4.1 – Comparison of the proposed approaches on the Raiconi and Gentili set of instances.

In general, it is observed that the CMLP-Greedy is a fast approach which runs in a low computational time, but it can find an optimal solution only for 7 out of 25 instances. An improvement of this method is proposed by the authors in which the previous constructive heuristic is embedded into a GRASP procedure used to solve the whole problem. Although the method performance is improved, an optimal solution is found for 60% of the experiments, at the cost of a significant increase of computational time.

As observed in Table 4.1, CG-EXACT is able to find the optimal solution for all the instances. Furthermore, the method is shown to be fast by solving all the problems in an average computational time of 12.7 seconds, which significantly outperforms previous approaches. However, as could be observed in Table 4.1, this lower computational time is the result of the good performance offered by the CG-MULTI approach, which is able to find all the optimal solutions in an average time of 7.47 seconds. As a matter of fact, with all the proposed instances, ILP is only used to prove optimality at the last iteration of the CG. This shows that the combination of GRASP and VNS is very efficient to find profitable columns. In addition, it can be seen that the proof of optimality can be obtained using ILP at the cost of a computational time increase of 41%.

CG-VNS is shown to be very efficient as it found the optimal solution for 23 out of 25 instances. The produced solutions have an average deviation of 0.06%. Nonetheless, it is observed that the time consumed by this approach is more than for the CG-MULTI approach. In contrast, CG-GRASP is shown to run in the smallest CPU times at the expense of solution quality. CG-GRASP is able to find the optimal solution for 60% of the instances and presents an average deviation to optimality of 2.8%.

The results in Table 4.1 show that optimal lifetime often takes an integer value. Two main reasons can explain why this happens, the number of sensors located one hop away from the base station and the maximum number of sensors covering each target. In the former case, an upper bound for network lifetime can be computed as the sum of the lifetime of each one of the sensors within the communication range of the base station. In the latter case, specially notorious in problems demanding full coverage of targets ($\alpha = 1$), the lifetime is limited by some critical targets that are covered by a few sensors.

A second group of instances is presented to evaluate the performance of the proposed approaches with different communication and sensing ranges. As proposed by Deschinkel [Deschinkel, 2011] and Singh *et al.* in [Singh et al., 2013], it is assumed that sensors and targets are randomly deployed in a 500×500 square area. A group of five sets of instances with $|S| \in \{100, 200, 300, 400, 500\}$ sensors are used. Two sets of targets $|K| \in \{15, 30\}$ and a randomly located base station are also given for each instance. The sensors are assumed to be identical and $b_{s_v} = 1 \forall s_v \in \mathcal{S}$. Four instances are generated for each combination of the previous parameters. Three levels of coverage are considered $\alpha \in \{0.7, 0.85, 1\}$. Finally, variations on the ratio between sensing and communication ranges are considered. For all the experiments, the communication range is $R_c = 125$, the sensing range R_s is in the set $\{100, 125\}$. This represents two sets of 120 instances whose solutions are presented in Tables 4.2 and 4.3. A time limit of 3600 seconds is established for all the experiments, if no optimal solution is found before the time limit, then the best solution found so far is reported.

Tables 4.2 and 4.3 present a comparison between the approaches proposed in this article to solve α -CMLP with different ratios between the sensing and communication ranges. The methods are compared in terms of the average objective function value, the average computational time and the number of optimal solutions found for each instance group. The tables also present separately the experimental results concerning the instances for which a proven optimal solution is known, as well as those for which the

optimal solution was not obtained before the time limit. Of course, only CG-EXACT is able to prove optimality.

S	T	α	CG-Exact			CG-MULTI			CG-VNS			CG-GRASP		
			LT	Av. Time(s)	#Opt	LT	Av. Time(s)	#Opt	LT	Av. Time(s)	#Opt	LT	Av. Time(s)	#Opt
100	15	0.7	6.88	8.78	4	6.63	4.08	3	6.88	4.97	4	5.60	0.76	1
		0.85	6.64	921.47	4	6.04	10.16	1	6.02	25.27	3	5.16	1.43	0
		1	4.00	1.80	4	4.00	0.84	4	2.50	0.85	2	3.79	0.32	1
100	30	0.7	7.00	6.54	4	6.75	4.75	3	7.00	5.35	4	5.96	1.57	0
		0.85	6.57	1922.59	2	6.49	44.82	2	6.29	82.50	1	5.28	2.22	0
		1	4.00	4.06	4	4.00	2.75	4	3.66	5.54	2	3.66	0.72	1
200	15	0.7	16.25	414.98	4	15.96	212.26	2	16.22	293.80	3	11.21	1.47	0
		0.85	15.42	941.60	3	15.37	253.00	3	15.30	496.17	3	10.82	2.82	0
		1	10.25	12.69	4	10.25	8.64	4	10.25	25.70	4	9.15	1.34	2
200	30	0.7	16.25	128.92	4	16.00	100.17	3	15.95	170.52	3	12.45	3.48	0
		0.85	15.35	1514.62	3	14.69	715.21	1	14.51	1365.67	1	11.76	7.92	0
		1	8.75	19.80	4	8.50	3.29	3	8.75	24.78	4	8.50	2.75	3
300	15	0.7	18.25	34.72	4	18.00	9.00	3	17.55	31.30	2	12.95	1.67	1
		0.85	18.25	91.10	4	18.00	28.88	3	18.00	83.05	2	12.56	3.79	0
		1	15.00	86.26	4	15.00	54.26	4	15.00	82.69	4	11.50	4.51	3
300	30	0.7	18.25	47.93	4	17.00	9.26	3	18.25	28.94	4	13.75	2.18	2
		0.85	18.25	104.34	4	16.44	39.28	3	16.25	116.98	3	13.58	11.35	1
		1	13.25	48.34	4	13.25	29.72	4	13.25	66.99	4	11.50	6.26	3
400	15	0.7	30.68	999.22	3	30.56	693.82	3	30.81	920.56	2	20.46	6.38	0
		0.85	28.66	1151.16	3	28.63	1077.91	3	28.64	1297.94	3	19.94	12.67	1
		1	18.25	95.70	4	18.25	28.67	4	18.25	68.89	4	16.34	5.58	0
400	30	0.7	29.55	1007.00	3	29.55	947.06	3	29.43	1013.43	3	25.72	32.54	1
		0.85	26.90	1907.09	2	26.90	1877.53	2	26.20	1886.35	2	23.08	92.89	1
		1	18.00	125.38	4	18.00	46.64	4	18.00	190.97	4	17.53	19.17	1
500	15	0.7	45.04	2554.30	2	45.04	2498.70	2	44.09	2252.47	2	32.65	34.75	0
		0.85	39.20	2742.29	1	39.20	2714.57	1	38.02	3145.49	1	30.45	74.19	0
		1	29.00	335.83	4	29.00	227.92	4	29.00	522.13	4	26.21	64.71	0
500	30	0.7	44.83	2748.99	1	44.83	2692.89	1	44.47	2661.04	1	37.00	98.20	1
		0.85	37.24	2768.34	1	37.24	2740.50	1	36.83	2870.65	1	32.35	190.26	0
		1	26.25	308.25	4	26.25	198.10	4	26.25	573.85	4	26.00	140.29	2
Av. Time (Optimal)			195.43			102.86			129.114			13.99		
Av. Time (Non Optimal)			3600			1724.43			1955.934			31.19		
Av. Time			768.47			575.82			677.16			27.61		
#Optimal solutions (%Opt)			100 (83.3%)			85 (70.8%)			84 (70.0%)			25 (20.8%)		
Av. LT Variation			-			1.61%			3.10%			18.40%		

Table 4.2 – Comparison of average lifetime and average running time ($R_s = 100$, $R_c = 125$).

The results confirm the observations obtained with the set of instances proposed by Raiconi and Gentili. CG-EXACT was able to find the optimal solution in 83.3% and 85% of the instances when $R_s < R_c$ and $R_s = R_c$ respectively. However, the experiments show that the performance decreases when instance size grows, as the number of iterations also increases. As a result, the CPU time for addressing ILP also increases. As shown in Tables 4.2 and 4.3, this behavior is accentuated when partial coverage is allowed ($\alpha < 1$).

Figure 3 presents the typical evolution of the CG-EXACT algorithm in terms of both, the objective function along CG iterations, and the approaches used to find an interesting column to be returned to RMP (see Figure 4.1). The phases are represented by bars and the height is divided in three levels, the first level indicates that the GRASP was successful in finding profitable columns for RMP, the second level indicates that GRASP failed and VNS was required. Finally, if VNS fails, the third level indicates that an iteration with ILP was performed.

The results confirm that ILP is mainly used at the last iteration of CG-EXACT, mostly with the purpose to prove optimality of the current solution. Experiments show that the use of ILP is only required 1.28 times per problem on average, and for 2.95% of iterations. By contrast, as presented in Figure 4.6, it accounts for 54.5% of the total CPU time of the CG-EXACT approach; nonetheless, as presented in Tables 4.2 and

S	T	α	CG-Exact			CG-MULTI			CG-VNS			CG-GRASP		
			LT	Av. Time(s)	#Opt	LT	Av. Time(s)	#Opt	LT	Av. Time(s)	#Opt	LT	Av. Time(s)	#Opt
100	15	0.7	7.00	1.79	4	7.00	0.84	4	7.00	4.01	3	6.66	0.51	2
		0.85	6.88	8.37	4	6.88	4.31	4	6.85	5.96	3	6.27	0.96	0
		1	4.75	3.62	4	4.75	2.07	4	4.50	5.95	4	4.41	0.51	2
100	30	0.7	7.00	3.05	4	6.75	0.68	3	6.75	10.07	4	6.75	0.67	3
		0.85	6.79	327.39	4	6.78	9.56	3	6.76	37.04	3	6.21	1.93	1
		1	4.75	91.41	4	4.71	4.94	3	4.60	3.59	3	4.40	0.96	2
200	15	0.7	16.25	19.56	4	16.00	8.88	3	16.25	128.62	3	12.41	0.90	0
		0.85	15.75	31.46	4	15.00	19.70	2	14.25	52.37	4	12.08	1.18	0
		1	13.00	46.56	4	12.50	36.06	3	12.41	66.94	4	11.27	2.49	1
200	30	0.7	16.25	19.34	4	16.25	6.61	4	15.66	151.76	4	13.65	1.83	0
		0.85	16.09	925.98	3	15.74	210.15	2	15.37	338.87	3	13.29	3.41	0
		1	11.75	96.81	4	11.75	96.65	4	11.75	100.81	4	10.75	3.71	1
300	15	0.7	18.25	27.97	4	18.25	6.22	4	18.00	42.00	4	13.50	0.65	2
		0.85	18.25	40.07	4	18.25	13.22	4	18.25	24.42	4	13.50	2.06	2
		1	16.75	52.10	4	16.75	23.10	4	16.75	28.17	3	13.00	3.44	2
300	30	0.7	18.25	29.23	4	18.25	6.59	4	18.25	92.93	4	15.25	1.47	2
		0.85	18.25	50.03	4	18.25	18.23	4	18.25	34.28	4	15.25	3.98	2
		1	16.00	82.83	4	15.75	27.25	3	16.00	53.17	4	13.91	7.95	1
400	15	0.7	33.36	991.85	3	32.86	908.98	2	33.09	884.22	4	24.40	5.83	1
		0.85	31.06	998.19	3	31.06	939.45	3	30.94	1330.59	3	23.42	12.17	1
		1	24.88	298.96	4	24.88	208.49	4	24.88	642.52	4	20.10	10.29	0
400	30	0.7	32.08	956.19	3	31.76	748.26	2	31.05	771.21	2	28.21	17.53	1
		0.85	29.08	1021.31	3	29.08	961.71	3	29.12	1043.68	3	24.84	27.10	1
		1	22.38	245.22	4	22.38	149.90	4	22.38	1212.56	4	20.42	23.53	1
500	15	0.7	50.51	1957.14	2	50.51	1901.83	2	47.71	1809.20	0	35.70	21.24	0
		0.85	45.58	2624.69	2	45.58	2569.32	2	37.82	2911.82	1	32.02	20.96	1
		1	36.82	1937.38	2	36.82	1881.92	2	36.79	3305.68	0	29.87	49.82	0
500	30	0.7	51.13	1947.69	2	50.51	1892.06	2	50.03	1720.43	1	39.58	38.96	1
		0.85	43.07	2741.61	1	43.07	2713.86	1	42.64	2723.56	1	35.02	72.01	0
		1	34.47	1938.50	2	34.47	1883.25	2	32.73	1967.87	2	30.68	136.32	0
Av. Time (Optimal)			128.5			73.69			103.9			6.51		
Av. Time (Non Optimal)			3600			2078.68			2555.54			19.8		
Av. Time			650.54			574.94			716.81			15.81		
#Optimal solutions (%Opt)			102 (85%)			90 (75%)			90 (75%)			25 (20.8%)		
Av. LT Variation			-			0.70%			2.40%			16.60%		

Table 4.3 – Comparison of average lifetime and average running time ($R_s = 125$, $R_c = 125$).

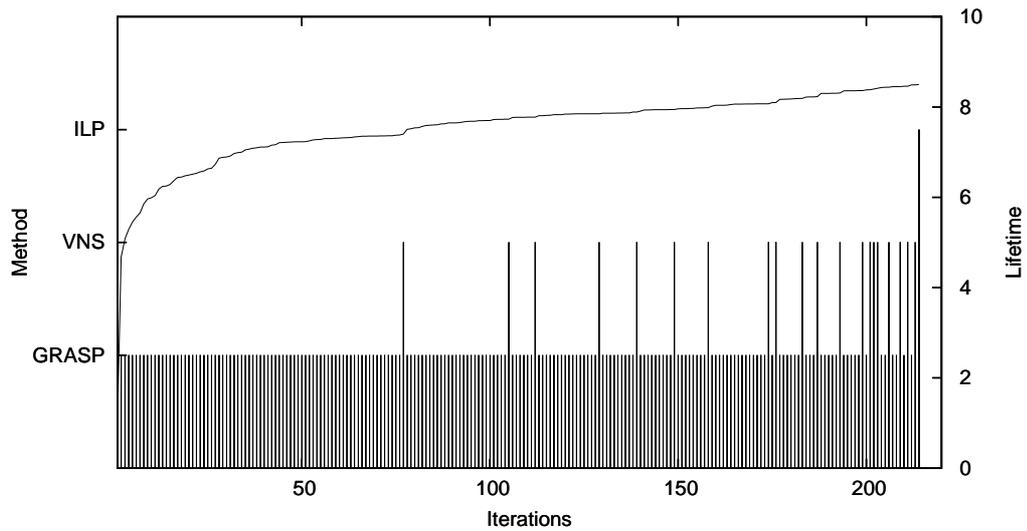


Figure 4.5 – Evolution of the CG-EXACT (Instance S100_T15_a70) .

4.3, the overall effect over the solutions quality is modest. In fact, the lifetime of the solutions returned by CG-MULTI, which is similar to CG-EXACT up to the moment where VNS fails to find any profitable cover, are only 1.15% less than with CG-EXACT.

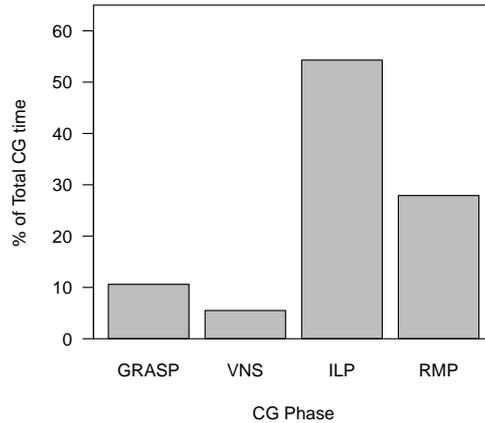


Figure 4.6 – Comparison of average time spent at each CG phase.

Regarding the contribution of VNS, it is shown to provide a fast and efficient alternative to GRASP for finding profitable columns. Then, VNS empowers the CG process to continue the search of interesting columns and take advantage of the low computational time consumed by the GRASP phase until it is unable to find interesting columns. Figure 4.5, shows the benefit of using VNS to increase lifetime when GRASP fails. VNS is used on 13.7% of the iterations and it only consumes 5.5% of the CPU time required by the CG-EXACT approach (see Figure 4.6). As expected, the VNS phase consumes a bit more of computational time than the GRASP. This is clearly explained by the fact that, during the implicit local search procedures, moves are only performed in the set of feasible solutions. It means that for each interesting neighbor of a solution for PS, connectivity and coverage must be checked.

Tables 4.2 and 4.3 show that CG-VNS can be used efficiently to solve α -CMLP. Nonetheless, by combining it with GRASP into CG-MULTI it is possible to reduce the computational time required to obtain optimal solutions. This conclusion is especially true for the instances that require the full coverage of targets ($\alpha = 1$). Of course, this can be partially explained because of GRASP procedure is expected to be more effective finding connected structures in problems with higher levels of coverage.

The results confirm the efficiency of using the GRASP approach to solve PS and return interesting columns for RMP. As observed in Figure 4.6, the GRASP phase consumes on average only 10.6% of the total CPU time required to solve α -CMLP. By contrast, the structure of the approach implies that it is used at each iteration. This fact means that, on average, the time spent generating interesting columns by using the GRASP and VNS approaches is lower than the total time used to solve the

RMP along the CG approach, which consumes on average the 27.9% of the total time. Although GRASP often fails to find an optimal solution for PS it is able to retrieve useful columns in a very low computational time, which suggests to use this approach for very large instances. Moreover, even when it fails, the GRASP method is still useful as a subprocess of CG-MULTI and CG-EXACT to keep the process running in a low computational time.

4.5 Conclusions and future work

In this chapter we have addressed the maximum network lifetime problem in wireless sensor networks with coverage and connectivity constraints. An extension of the mathematical model presented by Raiconi and Gentili [Gentili and Raiconi, 2013] is introduced to allow partial target coverage. An exact column generation approach is proposed to solve the problem. In order to speed up the method, it is empowered by the use of heuristic approaches to solve the pricing subproblem. A multiphase metaheuristic approach is introduced to help the CG framework to solve the problem in a low computational time. The method sequentially applies a GRASP and a VNS heuristic to find profitable solutions for PS and only applies ILP when both of them fail to find an interesting column. Furthermore, the exact algorithm is turned into a pure heuristic approach by turning off the ILP phase.

Experimental results confirm that the exact approach is efficient to solve the problem. Moreover, CG-MULTI is a very efficient heuristic, and emerges as promising candidate for addressing large and difficult problem instances. Although the proposed GRASP alone is not efficient for computing the optimal columns for PS along CG iterations, experiments confirm that it helps reducing the CPU time required to solve α -CMLP. Several reasons can be attributed to this result. First, the proposed GRASP procedure runs in low CPU times, and during the first iterations of CG it is easy to find columns with interesting reduced costs. As a consequence, the method allows to approach efficiently interesting solutions for α -CMLP. Moreover, by default the proposed GRASP procedure produces solutions in a diversified manner. In this way, the method enriches the RMP with columns contributing to different constraints. Then, an overall effect of reducing the number of iterations required to reach the optimal solution is observed.

Since α -CMLP may be enriched with additional constraints (like bandwidth constraints, adjustable sensing ranges), or with more complex energy consumption models, the proposed algorithms may serve as a basis to address these problems. More specifically, further research will consider the effect of having different roles and different energy consumption rates for the sensors. Additionally, the effect of the distance between sensors and between sensors and targets on the energy consumed by transmission and detection respectively will be considered.

5

Exact approaches for lifetime maximization in connectivity constrained wireless multi-role sensor networks

In this chapter, we consider the duty scheduling of sensor activities in wireless sensor networks to maximize the lifetime. We address full target coverage problems contemplating sensors used for sensing data and transmit it to the base station through multi-hop communication as well as sensors used only for communication purposes. Subsets of sensors (also called covers) are generated. Those covers are able to satisfy the coverage requirements as well as the connection to the base station. Thus, maximum lifetime can be obtained by identifying the optimal covers and allocate them an operation time. The problem is solved through a Column Generation approach decomposed in a master problem used to allocate the optimal time interval during which covers are used and in a pricing subproblem used to identify the covers leading to maximum lifetime. Additionally, Branch-and-Cut based on Benders' decomposition and Constraint Programming approaches are used to solve the pricing subproblem. The approach is tested on randomly generated instances. The computational results demonstrate the efficiency of the proposed approach to solve the maximum network lifetime problem in wireless sensor networks with up to 500 sensors.

5.1 Introduction

Wireless sensor networks (WSN) have undergone a growing popularity during the last decade [Arampatzis et al., 2005, Diamond and Ceruti, 2007, Hadjidj et al., 2013, Othman and Shazali, 2012]. The wide range of potential applications for sensors made them an interesting area of research [Yick et al., 2008, Zorbas and Douligeris, 2010]. Data such as temperature, light, sound or pressure can be collected by sensors and then transmitted to the user through multi-hop communication. Military applications as depot monitoring or intrusion detections in remote environments, industrial applications as inventory control, environmental monitoring, healthcare monitoring, among other fields are only a small sample of the fields into which WSN are used.

WSN are typically composed by a large amount of sensor nodes deployed to accomplish some monitoring and communications tasks. Sensors are constrained devices with

low computing capabilities that are basically composed by three components [Anastasi et al., 2009]: a sensing subsystem, a processing subsystem and a wireless communication subsystem. These components are coupled to guarantee that each device is able to collect information from the environment, to decide how to manage that information and how and where to transmit that information to be processed. Additionally, the power supply is obtained from a battery provided with a limited amount of energy. As a consequence, energy usage is a major concern in wireless sensor network design. In most applications, the use of wireless sensors demands the efficient design of strategies to manage their energy whilst keeping network operating properly.

Some applications require the sensors to be located in remote or hostile environments in which sensor placement cannot be controlled. Instead, sensors are randomly deployed from a remote location and their operations cannot be planned before their deployment. Hence, the replacement of sensors batteries is not possible. Consequently, some scenarios require the operation of sensors to answer to unknown operating conditions and topologies. Then, the way in which sensors are used must be defined *in situ* after network topology is known.

This research considers target coverage with wireless sensor networks, *i.e.*, sensors are used to collect information from the targets located within their sensing range. In order to efficiently use sensors battery's energy, a typical approach is to deploy more sensors than actually needed. Then, it is possible to identify subsets of sensors (covers) able to satisfy the coverage (*i.e.*, the coverage of some or all the targets) and connectivity requirements (*i.e.*, the information must be transmitted to a base station) [Deschinkel, 2011, Raiconi and Gentili, 2011, Rossi et al., 2012a, Zorbas et al., 2010]. Hence, lifetime, defined as the total time during which the WSN is able to provide target coverage and to send sensing information to the base station, is extended by activating these subsets at different moments. Therefore, such an approach can be successfully extended to consider WSN in which sensors can adopt different roles at different energy consumption rates.

A wide range of exact and heuristic approaches has been proposed to efficiently use the energy in wireless sensor networks. A complete review of approaches to efficiently use the energy on WSN is presented by Zorbas *et al.* [Zorbas and Douligieris, 2010]. Efforts have been mostly focused on the design of methods to maximize network lifetime by using heuristic criteria [Deschinkel, 2011, Gentili and Raiconi, 2013] and hybrid approaches as linear programming based rounding methods [Cardei et al., 2005]. Gu *et al.* [Gu et al., 2011] demonstrate that the solutions for the sensors coverage and scheduling problem can be accurately represented by using pattern structures, where patterns (covers) represent the energy consumption rate of sensors during the time interval in which they are active. Exact approaches based on column generation (CG) are currently state-of-the-art algorithms to solve coverage and scheduling problems in wireless sensor networks [Alfieri et al., 2007, Castaño et al., 2013, Gu et al., 2009b, Rossi et al., 2012a, Singh et al., 2013]. CG relies on covers and decomposes the problem in two subproblems. The restricted master problem (RMP) maximizes the network lifetime on a restricted set of covers, and the pricing subproblem (PS) generates new covers that may increase lifetime even further in the master problem.

CG has been shown to be efficient when simple network models are considered,

however addressing the subproblem becomes very challenging when network models get more complex, *e.g.*, when connectivity is required or the adoption of different roles for sensors within the network is allowed. Hence, improvements to the classical exact methods are necessary in order to guarantee that optimal solutions are obtained.

Most research on maximum network lifetime in WSN is concerned with the design of strategies to efficiently use the energy considering a homogeneous set of sensors being either active or inactive [Cardei and Cardei, 2008a, Cardei et al., 2005, Gentili and Raiconi, 2013, Lu et al., 2005, Raiconi and Gentili, 2011, Slijepcevic and Potkonjak, 2002]. An active sensor is able to monitor all the targets that are located within its sensing range and to establish communication with other sensors, or the base station, if they are located within its communication range. A sensor is *inactive* if it is neither monitoring nor transmitting and it operates at negligible energy consumption rate without performing any activity within the network. In this chapter we adopt a more general approach to consider wireless sensors having up to three operation modes [Zhao and Gurusamy, 2008]. An active sensor is a *source* if it is performing monitoring and transmission tasks and is a *relay* if it is only used to transmit the information collected by source nodes to other sensors or to the base station. We will show that both scenarios can be tackled in a similar way since the case considering only two operation modes is a special case of the previous one.

In order to solve the problem we propose an exact CG approach, where two methods are proposed for addressing the pricing subproblem. The first method used to address the pricing subproblem, also decomposes it into two, that are solved through a Branch-and-Cut method based on Benders' decomposition. Additionally, the method is reinforced by adding some valid inequalities and connectivity cuts that help to improve the performance of this approach. The second method used to address the pricing subproblem is based on constraint programming. It uses specialized graph variables with global constraints (tree constraint, globalCardinality constraint and channeling constraints) to model an underlying pricing subproblem. A specialized constraint propagation scheme guided by specific problem information is used to bring results in a quasilinear number of decisions throughout the search process.

This chapter is structured as follows. In Section 5.2 a detailed description of the maximum network lifetime problem with connectivity constraints is provided as well as the mathematical model adopted to tackle this problem. In Section 5.3, the solution approach based on CG adopted to solve the problem is presented. Section 5.4 introduces the two algorithms for addressing the pricing subproblem. In Section 5.5, the performances of these algorithms are measured, by incorporating them into the CG framework, and by solving the lifetime maximization problem on a large set of instances. Finally, conclusions and future work are presented in Section 5.6.

5.2 Problem description

Consider a set of sensors $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$ randomly deployed to monitor a set of targets $\mathcal{K} = \{k_1, k_2, \dots, k_n\}$ and to transfer sensing information to the base station,

denoted by S_0 . Each sensor is able to assume three different roles within the network. A sensor is a *source* node if it collects information from the targets that are located within its sensing range R_s and transfer information to the base station or to other nodes located within its communication range R_c . In addition to its own collected data, a source may also transfer information sent by other sensors. A sensor is a *relay* node if it is not sensing and is used only to re-transmit information collected by source nodes to other sensors or to the base station. Finally, a sensor that is neither collecting nor transferring information is *inactive*. The complete list with the notation used along the chapter is summarized in Table 5.1.

Notation	Meaning
m	Number of sensors
n	Number of targets
\mathcal{S}	Set of sensors
\mathcal{K}	Set of targets
S_0	Base station
R_s	Sensing range
R_c	Communication range
E_s	Energy consumption rate for a sensor operating as a source
E_r	Energy consumption rate for a sensor operating as a relay
E_i	Energy consumption rate for an inactive sensor
\mathcal{E}	Set of energy consumption rates $\mathcal{E} = \{E_s, E_r, E_i\}$
S_s^j	Set of sensors operating as sources
S_r^j	Set of sensors operating as relays
S_i^j	Set of inactive sensors
P_j	Partition of \mathcal{S} into S_s^j , S_r^j and S_i^j
Ω	Set of all the feasible partitions that satisfy the coverage and connectivity requirements
\mathcal{N}	Set of nodes equal to $\mathcal{S} \cup \mathcal{K} \cup \{S_0\}$
\mathcal{N}_j	Subset of nodes $\mathcal{N}_j \leftarrow \mathcal{N} \setminus S_i$
\mathcal{A}	Set of existing arcs between the nodes of \mathcal{N}
$G(\mathcal{N}, \mathcal{A})$	Graph containing the whole set of nodes and arcs
$G[\mathcal{N}_j]$	Subgraph of G induced by \mathcal{N}_j
t_j	Decision variables that denotes the amount of time allocated to partition P_j
b_{s_u}	Battery's capacity of the sensor s_u
x_{uv}	Flow passing between nodes $u, v \in \mathcal{N}$
$y_{s_v \ell_j}$	Binary role allocation variable for sensor s_v in partition P_j
$T[\mathcal{N}^T, \mathcal{A}^T]$	Directed graph variable for Constraint Programming model

Table 5.1 – Notations.

Let $G(\mathcal{N}, \mathcal{A})$ be a directed graph where $\mathcal{N} = \mathcal{S} \cup \mathcal{K} \cup S_0$ is the set of nodes and \mathcal{A} is the set of arcs used to indicate if communication is possible between sensor nodes or if a target is monitored by a given sensor. An arc $a(u, v) \in \mathcal{A}$, used to represent the possibility of transmitting information between the elements of the network, exists if: (i) $u \in \mathcal{K}$, $v \in \mathcal{S}$ and u is located within the sensing range R_s of v , (ii) $u, v \in \mathcal{S}$ and they are located within the communication range of each other or (iii) if the base station $v = S_0$ is located within the communication range of a sensor $u \in \mathcal{S}$.

The energy consumption rate (*i.e.*, the consumed power) of a sensor only depends upon its current role: the consumption rate of a source (respectively a relay and an inactive sensor) is denoted by E_s (respectively E_r and E_i). The set $\mathcal{E} = \{E_s, E_r, E_i\}$ contains the energy consumption rates of the sensors. In general $E_s > E_r > E_i$, but this hypothesis is not necessary here. If $E_s \leq E_r$, the relay mode is useless and can be discarded, consequently an active sensor is always a source. The latter case has been

considered by Castano *et al.* [Castaño et al., 2013], where the use of hybrid approaches is proposed to solve the problem.

Let P_j be a partition of \mathcal{S} into three (non overlapping) sets: the source nodes (S_s^j), the relay nodes (S_r^j) and the inactive nodes (S_i^j). We define \mathcal{N}_j as $\mathcal{N}_j = \mathcal{N} \setminus S_i^j$ and let $G[\mathcal{N}_j]$ denote the subgraph of G induced by \mathcal{N}_j . The partition P_j is said to be feasible if for all targets $k_i \in \mathcal{K}$, there exists a path from k_i to S_0 in $G[\mathcal{N}_j]$. Let Ω denote the set of all feasible partitions P_j , the maximum network lifetime problem with role allocation and connectivity constraints (CMLP-MR) is to find the optimal utilization time of these partitions; so as to maximize lifetime while respecting the battery's capacity b_{s_v} of the sensors. The case in which $E_s = E_r$ is known as connected maximum network lifetime problem (CMLP)[Castaño et al., 2013, 2014, Raiconi and Gentili, 2011].

The set of binary variables $y_{s_u \ell_j}$ is used to identify the structure of the partition P_j . It is used to indicate whether or not a sensor s_v assumes the energy consumption profile $\ell \in \mathcal{E}$ in feasible partition $P_j \in \Omega$. In this way, a cover is defined as a vector indicating the power allocated to each particular sensor according to the partition P_j . Hence, the structure of the cover can be derived from the structure of the partition. Figure 5.1 presents a simple representation of the concept of partition. In Figure 5.1a a potential network configuration is presented, where dotted lines indicate the possible links between the sensors (circles) and the base station (triangle) or the sensors and the targets (square). In Figures 5.1b and 5.1c two different feasible partitions of \mathcal{S} are represented; the solid lines indicate the connections actually established between the nodes of the network. These partitions correspond to $S_s^j = \{s_5, s_7\}$, $S_r^j = \{s_1, s_3\}$ and $S_i^j = \{s_2, s_4, s_6\}$ for Figure 5.1b and $S_s^j = \{s_4, s_6\}$, $S_r^j = \{s_1, s_2\}$ and $S_i^j = \{s_3, s_5, s_7\}$ for Figure 5.1c. Finally, Figure 5.1d presents an infeasible partition that cannot provide full target coverage.

As previously mentioned, the objective is to maximize network lifetime by allocating an optimal time interval t_j for each partition P_j to be used. For each sensor, a battery constraint is imposed to guarantee that the allocated time intervals respect the limited energy resources considering that sensors can operate at different energy consumption rates. Then, the implicit network configurations are represented by columns (covers) indicating at each row the energy consumption rate at which their nodes operate for each time interval along network lifetime. Assuming that the set Ω is completely known, and so the values $y_{s_u \ell_j}$, it is possible to formulate the problem by using the following linear programming model (Model M1):

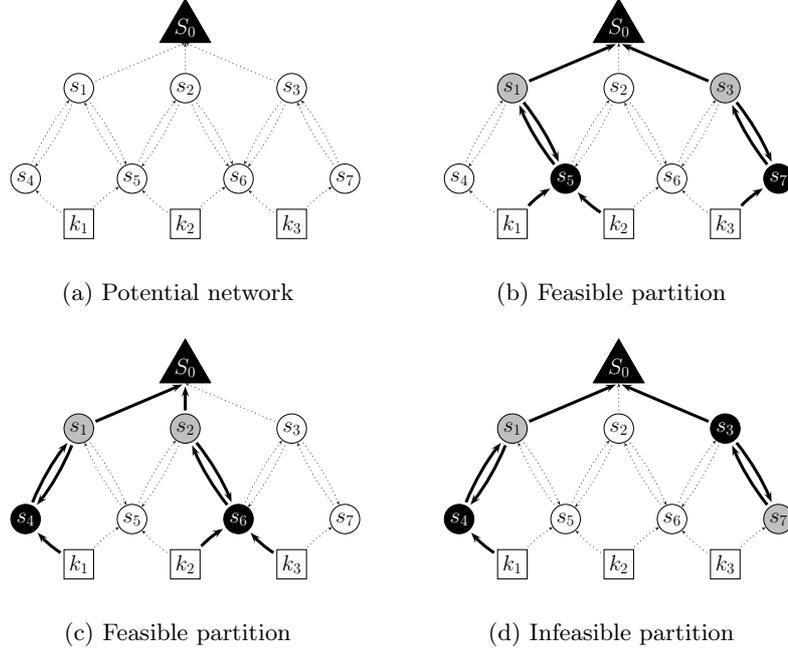


Figure 5.1 – Representation of feasible and infeasible partitions in a simple network.

Model M1: Maximum network lifetime problem

$$\text{Maximize : } \sum_{j|P_j \in \Omega} t_j \tag{5.1}$$

$$\sum_{j|P_j \in \Omega} (\sum_{\ell \in \mathcal{E}} \ell y_{s_u \ell j}) t_j \leq b_{s_u} \quad \forall s_u \in S \tag{5.2}$$

$$t_j \geq 0 \quad \forall j|P_j \in \Omega \tag{5.3}$$

The objective of model (5.1-5.3) is to maximize network lifetime (5.1) by using the partitions $P_j \in \Omega$. Constraints (5.2) are used to guarantee that the initial amount of energy b_{s_u} for each particular sensor is respected. As expected, time variables t_j are continuous (5.3), and can be upper-bounded by taking into account initial battery level and consumption rate. However, enforcing these upper bounds is of no use for solving this problem.

5.3 Solution approach

In Model M1, the number of columns grows exponentially with the number of sensors. Consequently, assuming that the whole set of feasible partitions is known in advance

is not always realistic. Moreover, it is well known that, in the optimal solution, only part of these columns will be used for a non zero time, and the remaining will not be useful at all. Therefore, a more intelligent strategy is to generate them only if they are required and promising to extend network lifetime. In this chapter the use of CG is proposed as a strategy to identify interesting partitions leading to network lifetime improvements. Considering the fact that Model M1 is a LP problem that can be easily solved with standard solvers when the number of columns is not prohibitively large, the focus is to efficiently generate those interesting connected structures, called attractive covers.

In Model M1, the CG algorithm starts with a subset of covers $\Omega' \subseteq \Omega$. Solving M1 on that subset allows to access optimal dual variables values π_v associated to each sensors' battery constraint. Then, these values are used in the pricing subproblem (PS) to identify only interesting partitions leading to lifetime improvements. In other words, PS is used to gradually enlarge set Ω' with interesting covers that are added to the initial pool of columns of M1, allowing to increase the network lifetime. The reduced cost criterion is used to identify those interesting columns. If the optimal network configuration obtained by PS produces a column with positive reduced cost, which means that it has potential to extend lifetime, it is added to the set Ω' (*i.e.*, an additional column in the Model M1) and the process starts again. Otherwise, the CG process terminates, proving that the current solution of the master problem is optimal.

5.3.1 Pricing subproblem

Any valid cover derived from partition P_j has to cover all the targets. Consequently, in order to generate interesting covers, we devise a network flow model where each target sends one unit of flow. The coverage requirement is then satisfied by a cover if and only if the base station receives $|\mathcal{K}|$ units of flow. The variables x_{uv} are integer variables used to identify the amount of flow passing through the link $a(u, v)$ and the binary variable y_{s_v, ℓ_j} is used to identify the energy consumption profile assumed by sensor s_v . Such a situation is represented by using the following mathematical model:

 Model M2: Pricing subproblem

$$\text{Maximize : } 1 - \sum_{v \in \mathcal{S}} \sum_{\ell \in \mathcal{E}} \pi_v \ell y_{s_v \ell j} \quad (5.4)$$

Subject to:

$$\sum_{u \in \mathcal{S} | \exists a(v, u)} x_{vu} = 1 \quad \forall v \in \mathcal{K} \quad (5.5)$$

$$\sum_{u \in \mathcal{N} | \exists a(v, u)} x_{vu} - \sum_{u \in \mathcal{N} | \exists a(u, v)} x_{uv} = 0 \quad \forall v \in \mathcal{S} \quad (5.6)$$

$$\sum_{u \in \mathcal{N} | \exists a(u, S_0)} x_{uS_0} = |\mathcal{K}| \quad (5.7)$$

$$x_{uv} \leq |\mathcal{K}| (y_{s_v E_{rj}} + y_{s_v E_{sj}}) \quad \forall u, v \in \mathcal{S} | \exists a(u, v) \quad (5.8)$$

$$x_{vu} \leq |\mathcal{K}| (y_{s_v E_{rj}} + y_{s_v E_{sj}}) \quad \forall v, u \in \mathcal{S} | \exists a(v, u) \quad (5.9)$$

$$x_{uv} \leq y_{s_v E_{sj}} \quad \forall u \in \mathcal{K}, \forall v \in \mathcal{S} | \exists a(u, v) \quad (5.10)$$

$$\sum_{\ell \in \mathcal{E}} y_{s_v \ell j} = 1 \quad \forall v \in \mathcal{S} \quad (5.11)$$

$$x_{uv} \in \mathbb{Z}^+ \cup \{0\} \quad \forall u, v \in \mathcal{N} \quad (5.12)$$

$$y_{s_v \ell j} \in \{0, 1\} \quad \forall v \in \mathcal{N}, \ell \in \mathcal{E} \quad (5.13)$$

As previously mentioned, the objective function is to maximize the reduced cost (5.4). The equations (5.5-5.7) are balance flow conservation constraints used to guarantee that flow arise only on targets (5.5), passes through sensor nodes without any consumption (5.6) and is directed to the base station which “consumes” that flow (5.7). Constraints (5.8-5.9) guarantee that links between nodes of the network are used only if the sensors operating the corresponding sensing and transmission are either sources or relays. In the same way, constraints (5.10) allow the existence of sensing links only if the corresponding sensor is active as a source; only one unit of information is originated in the target node. Finally, constraint (5.11) is used to guarantee that sensors assume exactly one of the three energy consumption profiles presented in this chapter.

5.4 Solution approaches to address pricing subproblem

As it will be shown, the classical implementation of CG based on state-of-the-art solvers present several drawbacks that may affect the usability of CG to solve CMLP and CMLP-MR. Instead, in this chapter a Branch-and-Cut based on Benders’ decomposition (BBC) [Benders, 1962] approach, and a Constraint Programming approach are developed to efficiently solve PS.

5.4.1 A decomposition approach to address pricing subproblem

Consider the Model M2. If the values for the set of variables $y_{s_v \ell_j}$ are predefined, and considering that no cost is associated to the x_{uv} variables, the problem is reduced to check whether or not each target can send one unit of flow to the base station. Then, the set of variables $y_{s_v \ell_j}$ can be considered as a set of complicating variables increasing the difficulty of the PS. If their values are known, the number of integer variables required to obtain an optimal solution for PS is largely reduced as the configuration of the network becomes easy to compute.

Benders' decomposition [Benders, 1962] is an approach for exploiting the structure of mathematical programming problems with complicating variables [Geoffrion, 1972]; Benders' decomposition has been largely used in network design problems [de Camargo et al., 2008, de Sá et al., 2013, Easwaran and Üster, 2009]. As in column generation, Benders' decomposition divides a problem in two subproblems: A Benders master problem (BMP) containing only complicating variables and a Benders' subproblem (BSP) useful to create a correct representation of the optimal solution, based on the extreme directions and extreme points of the dual problem associated to the BSP.

In this chapter the proposal is to use BBC in order to solve PS, *i.e.*, to generate interesting columns for Model M1. On one hand BMP is used to allocate energy consumption rates to the sensors on the network depending on the task assigned to them in the network. On the second hand, BSP is used to check whether or not the solution found is feasible in terms of connectivity and coverage. When it is not feasible, it can be used to generate new cuts to be included in BMP and guide the $y_{s_v \ell_j}$ values to the optimal solution. The general approach adopted to solve CMLP-MR is depicted in Figure 5.2.

Let θ_v^d denote the dual variable values associated with flow constraints (5.5), and $\theta_{S_0}^d$ the dual variable value associated with flow constraint. β_{uv}^d and η_{uv}^d are used to identify the dual variables related with constraints (5.8-5.9) and (5.10) respectively. The set of extreme rays computed iteratively through BSP is denoted by \mathbb{D} . The Benders reformulation of Model M2 can be expressed as follows through Model M3:

 Model M3: Benders' Master Problem

$$\text{Maximize : } 1 - Z_{BMP} \quad (5.14)$$

Subject to:

$$Z_{BMP} \geq \sum_{v \in \mathcal{S}} \sum_{\ell \in \mathcal{E}} y_{s_v \ell j} \ell \pi_v \quad (5.15)$$

$$\begin{aligned} & \sum_{u, v \in \mathcal{S} | \exists a(u, v)} (y_{s_v E_{rj}} + y_{s_v E_{sj}}) \beta_{uv}^d |\mathcal{K}| + \sum_{v, u \in \mathcal{S} | \exists a(v, u)} (y_{s_v E_{rj}} + y_{s_v E_{sj}}) \beta_{vu}^d |\mathcal{K}| + \\ & \sum_{v \in \mathcal{K}} \theta_v^d - |\mathcal{K}| \theta_{S_0}^d + \sum_{v \in \mathcal{K}} \theta_v^d - |\mathcal{K}| \theta_{S_0}^d + \sum_{u \in \mathcal{K}, v \in \mathcal{S} | \exists a(u, v)} (y_{s_v E_{sj}}) \eta_{uv}^d \geq 0 \quad \forall d \in \mathbb{D} \end{aligned} \quad (5.16)$$

$$\sum_{\ell \in \mathcal{E}} y_{s_v \ell j} = 1 \quad \forall v \in \mathcal{S} \quad (5.17)$$

$$\sum_{v \in \mathcal{S} | \exists a(u, v)} y_{s_v E_{sj}} \geq 1 \quad \forall u \in \mathcal{K} \quad (5.18)$$

$$\sum_{v \in \mathcal{S} | \exists a(v, S_0)} (y_{s_v E_{rj}} + y_{s_v E_{sj}}) \geq 1 \quad (5.19)$$

$$\sum_{v | \exists a(u, v)} (y_{s_v E_{rj}} + y_{s_v E_{sj}}) \geq (y_{s_u E_{rj}} + y_{s_u E_{sj}}) \quad \forall u \in \mathcal{S} | \nexists a(u, S_0) \quad (5.20)$$

$$y_{s_v \ell j} \in \{0, 1\} \quad \forall v \in \mathcal{N}, \forall \ell \in \mathcal{E} \quad (5.21)$$

$$Z_{BMP} \geq 0 \quad (5.22)$$

As previously mentioned, the model above is used to identify interesting columns for Model M1 which is calculated based on the reduced cost criterion (5.14). The set of Benders feasibility cuts is represented by constraints (5.16), it consists in an exponential set of constraints that brought the information provided by the infeasible solutions found for BSP. Initially, these constraints are not included in BMP as they are added afterwards by computing Benders' feasibility cuts.

A typical solution for the pricing subproblem is highly affected by the set of variables $y_{s_v \ell j}$. As observed in Model M2, the allocation of roles to the sensors affects network structure as well as feasibility. Consequently, it is possible to define structural inequalities, based on these variables, which are valid for the model and help to strengthen the model formulation [Saharidis et al., 2011]. The set of constraints (5.18) are used to enforce coverage, *i.e.*, at least one sensor operating as a source must be active around each target. In the same way, constraint (5.19) indicates that at least one active sensor must be located within the communication range of the base station S_0 in order to transmit the collected flow to this one. Finally, the constraints (5.20) are used to forbid sensors operating as source or relay, that are not located within the communication range of the base station S_0 , to be active if they are isolated.

The classical implementations of Benders' decomposition implies that an IP problem has to be solved iteratively to prove optimality each time that a new cut is found. This

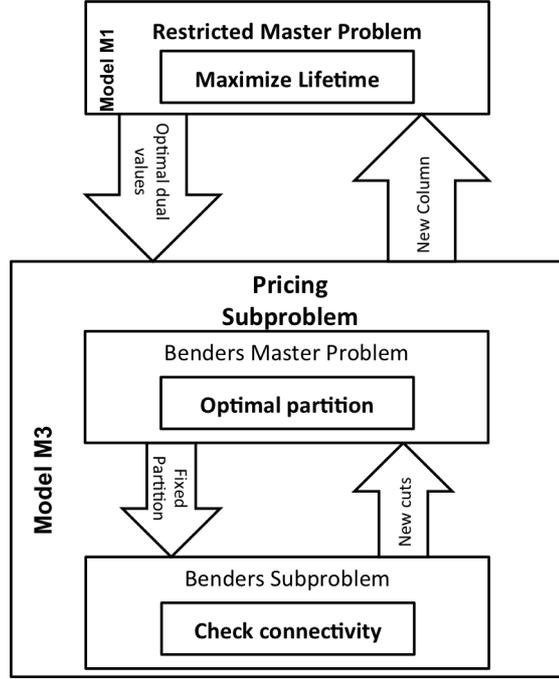


Figure 5.2 – Column generation approach to solve CMPL-MR

might not be a problem at the beginning of the solution process, when the number of constraints is low; nonetheless, it can be quite complex when new constraints are added. Moreover, no feasible solution is obtained until the end of the solution process. To overcome this problem, we embed the generation of Benders' feasibility cuts within a Branch-and-Cut strategy (BBC) as proposed by Errico *et al.* [Errico et al., 2012]. In this way, it is possible to take advantage of the infeasible integer solutions found to improve the solution process, and to reduce the computational time.

The new cuts for BMP are generated through BSP. Considering the fact that variables x_{uv} make no contribution to the objective function, the information leading to construct useful cuts is only derived from infeasible solutions for BSP, or the unbounded solution for the dual associated to BSP. The information required to compute these new cuts is obtained by computing the Farkas' dual variables associated to BSP when it is infeasible. Then, the classical constraint, widely known as Benders' feasibility cut as can be seen in Equation (5.16), is built using this information. Moreover, it is possible to construct an additional set of constraints to strengthen the BBC strategy. It is intended to avoid connected components that are not connected to the base station. As for the Benders' feasibility cuts, these new cuts are added to Model M3 each time that an infeasible solution for BSP is found.

Lemma 1 Consider an infeasible partition obtained through pricing subproblem. Let $\mathcal{C} \subseteq \mathcal{S}$ denote the set of connected components that do not contain the base station neither has any sensor able to communicate to this one and cc_i a connected component

in this set. Then, in order to recover feasibility, it is at least necessary for each connected component cc_i to establish a connection with a sensor out of it. That is, it must satisfy the following constraint:

$$\sum_{s_u \in cc_i} (y_{s_u E_r j} + y_{s_u E_s j}) \leq |cc_i| \sum_{\substack{s_v \in \mathcal{S} \setminus cc_i \\ |s_u \in cc_i, \exists a(u,v)}} (y_{s_v E_r j} + y_{s_v E_s j}) \quad \forall cc_i \in \mathcal{C} \quad (5.23)$$

Proof If a connected component cc_i not connected to the base station is required to monitor the targets, the information retrieved by the sensors in this set need to be transmitted to the base station through multi-hop communication. In other words, at least one sensor able to establish communication with the nodes in cc_i has to be active.

It is widely known that column generation can present some drawbacks that may limit its usability [Marsten et al., 1975, Mounpla et al., 2010, Vanderbeck, 2005]. Especially in WSN networks lifetime optimization, Gu *et al.* [Gu et al., 2009b] demonstrated that the use of approximated criterion might be necessary to stop the CG process when a near optimal solution is detected. In CMLP and CMLP-MR the use of state-of-the-art ILP solvers to solve PS tends to generate densely populated columns, *i.e.*, partitions with an unnecessary high number of active sensors, in those iterations in which most of the associated dual variables have a zero value. This is detrimental to CG convergence.

Several factors have been identified as the causes of the slow convergence of CG and a bunch of approaches to attack this problem have been proposed when this is an effect of the degeneracy problems in the master problem [Marsten et al., 1975]. However, it might not be the case in CMLP and CMLP-MR. Consequently, we propose the use of a different approach to speed up convergence. An additional constraint is used to limit the number of sensors active in a given column returned through PS and to avoid the *heading-in* effect observed during the first iterations. This constraint is modified dynamically throughout CG to guarantee that in the end the solution found is optimal. Let Max_Act_j be an arbitrarily selected upper bound on the maximum number of active sensors (as source or relays) in a partition, or column in the model M1. In order to avoid generate highly populated columns the following constraint can be added to Model M3:

$$\sum_{v \in \mathcal{S}} (y_{s_v E_r j} + y_{s_v E_s j}) \leq Max_Act_j \quad (5.24)$$

As expected, the previous constraint may be removing feasible solutions that are potentially interesting to continue the CG process. This means that, by considering this constraint, CG may converge prematurely, perhaps in a suboptimal solution if an interesting column is cut off by constraint (5.24). In order to overcome this issue and to keep the method operating as an exact approach, a dynamic modification of the value of Max_Act_j is proposed (Algorithm 1). The general idea is to start CG with a reduced search space and gradually enlarge this by a factor $(1 + \alpha)$, where $\alpha > 0$, throughout the CG process. Finally, once the value of Max_Act_j reaches m , the number of sensors

in the network, and assuming that no positive reduced cost columns exists, the current solution of the pricing subproblem is optimal.

Algorithm 5.1: Acceleration procedure for column generation

```

1  $j \leftarrow 0$ 
2  $RMP(\Omega'), Lifetime \leftarrow Initial\_solution();$ 
3  $P_j \leftarrow \emptyset;$ 
4  $Z_{BMP} \leftarrow 0$ 
5  $Act\_limit\_ctr \leftarrow True$ 
6 while  $Z_{BMP} > 0$  or  $Act\_limit\_ctr$  do
7    $j \leftarrow j + 1$ 
8   if  $Max\_Act_j \geq |S|$  then
9      $Act\_limit\_ctr \leftarrow False$ 
10  end
11   $\bar{\pi}, Lifetime \leftarrow Solve\_RMP(RMP)$ 
12   $P_j, Z_{BMP} \leftarrow Solve\_PS(\bar{\pi})$ 
13  if  $Z_{BMP} > 0$  then
14     $\Omega' \leftarrow \Omega' \cup P_j$ 
15  end
16  else
17    if  $Act\_limit\_ctr$  then
18       $Max\_Act_j \leftarrow Max\_Act_j \times (1 + \alpha)$ 
19    end
20  end
21 end

```

5.4.2 A constraint programming approach to address the pricing subproblem

We now propose a constraint programming approach to solve PS. In a first step, we describe the variables used in the model, then the constraints involved and finally a search strategy.

Variables We introduce a directed graph variable $T[\mathcal{N}^T, \mathcal{A}^T]$ [Dooms et al., 2005] based on the graph $G(\mathcal{N}, \mathcal{A})$, that will represent the underlying solution. \mathcal{N}^T is a subset of nodes ($\mathcal{N}^T \subseteq \mathcal{N}$) activated in the partition and \mathcal{A}^T is the subset of arcs ($\mathcal{A}^T \subseteq \mathcal{A}$) incident to \mathcal{N}^T . A mandatory set of nodes \mathcal{K} and the base station S_0 directly belong to the kernel of nodes set of this graph variable.

We manipulate also integer domain variables vectors of size m : *i*) $Succ$ where $D(Succ_u) = \{1, \dots, m\}$ represents the index of the successor of node u ($Succ$ is composed by two parts, successors from target node \mathcal{K} , called $SuccK$, and successors from sensor nodes \mathcal{S} , called $SuccS$), *ii*) $DegS$ and $DegK$ where $D(DegS_u) = \{0, \dots, m\}$ and $D(DegK_u) = \{0, \dots, n\}$ represent incoming degree on node $u \in \mathcal{S}$ respectively from \mathcal{S} and \mathcal{K} and *iii*) $Cost$ where $D(Cost_u) = \{0, \dots, \pi_u\}$ represent the associated dual value of node u . Finally an integer variable $TotalCost$ sum up all these $Cost$ variables (where variables are in upper case, as is common in constraint programming).

Constraints We have the following CP Model M4:

Model M4: Pricing Problem in Constraint Programming

$$tree(T, 1) \tag{5.25}$$

$$sumWeight(T, Cost, TotalCost) \tag{5.26}$$

$$TotalCost \leq 1 \tag{5.27}$$

$$channeling(T, Succ) \tag{5.28}$$

$$globalCardinality(SuccS, S, DegS) \tag{5.29}$$

$$globalCardinality(SuccK, S, DegK) \tag{5.30}$$

$$DegK(u) > 0 \iff Cost(u) = E_s \pi_u \quad \forall u \in \mathcal{S} \tag{5.31}$$

$$DegS(u) > 0 \implies Cost(u) \geq E_r \pi_u \quad \forall u \in \mathcal{S} \tag{5.32}$$

$$DegS(u) + DegK(u) = 0 \iff Cost(u) = E_i \pi_u \quad \forall u \in \mathcal{S} \tag{5.33}$$

Equations (5.25) imply that T is constrained to be a unique tree, as described by Lorca *et al.* [Fages and Lorca, 2011] and Unsworth [Prosser and Unsworth, 2006]. Complexity to maintain such a constraint is referred to be enforced in $O(|\mathcal{N}^T| + |\mathcal{A}^T|)$. Constraints (5.26-5.27), indicate that the overall sum of the cost associated to the nodes S , referred as $TotalCost$, is restricted to be less than 1 to ensure generation of valid columns for the master problem. Only bound consistency on $TotalCost$ is necessary and the associated scalar complexity is linear in $O(|\mathcal{N}^T|)$.

Constraints (5.28-5.33) are associated to the cost behavior. First, $Succ$ variables are derived from T via a standard channeling constraint (5.28) to maintain the domain of variable $Succ$ equivalent to the neighbors of nodes in T , *i.e.* $v \in D(Succ_u) \iff v \subset N(u)$ of T . Second, two global cardinality constraints are used to compute the occurrence number of node u in \mathcal{S} from respectively their neighbors of \mathcal{S} and \mathcal{K} (5.29-5.30), *i.e.*, incoming degree from \mathcal{S} and from \mathcal{K} . Finally, the three remaining constraints describe the three different energy consumption conditions of a node, with associated rates E_s , E_r and E_i .

Search Strategy As a search implementation, we use a standard max regret strategy on Cost variables. The regret here is the difference between the smallest value ($E_i \pi_u$) and the next value ($E_r \pi_u$) of the variables. The biggest regret (and first in enumeration) occurs for the most penalized nodes. Value assignment starts from minimum value (*i.e.* trying to deactivate the node). Reachability propagator inside the tree constraint ensures that the necessary connecting nodes will not be removed in the path between the targets and the base station in the final solution.

Equations (5.25) and (5.26) could be aggregated in a single global weighted node spanning tree constraint where propagator can be developed in the spirit of what has

been done for the weighted edge spanning tree constraint [Régim et al., 2010]. We decide to let a simple propagator because 94% of instances are solved optimally with an average complexity to find new column less than the total number of nodes in the graph (*i.e.*, propagation automatically fixes the last nodes and very few backtracks happen).

5.5 Computational experiments

In order to evaluate the advantages offered by each of the proposed approaches a set of experiments is conducted to compare the performance over a set of 160 instances. The instances were randomly generated with a number of sensors in $|\mathcal{S}| = \{100, 200, 300, 400, 500\}$ and a number of targets in $|\mathcal{K}| = \{15, 30\}$. Sensors are assumed to be deployed in a 500×500 area with a fixed communication range R_c of 125 and two different sensing ranges $R_s = 100$ and 125 [Castaño et al., 2014]. We assume that the energy consumption rate associated to a sensor acting as a source is $E_s = 1.0$, as a relay is $E_r = 0.8$, and inactive $E_i = 0$ for the case of CMLP-MR. By contrast, in CMLP it is assumed that $E_s = E_r = 1.0$ and $E_i = 0$.

A CPU time limit of 3600 seconds is established for the execution of the proposed approaches. The best solution found during this time is returned and used for comparison. The experiments were performed over a workstation with 6GB RAM DDR3 and an Intel Xeon Quad-Core W3550 processor @3.06GHz.

Tables 5.2 and 5.3 present the results obtained through the single application of an ILP solver to solve the pricing subproblem during CG iterations (CG+ILP), the proposed BBC strategy (CG+BBC) and the constraint programming approach (CG+CP). The columns m , n and R_s indicate respectively the number of sensors, targets and the sensing range of the instances. The column Opt/**BKS*** indicates the optimum or best known solution for the considered group of instances. In order to compare the performance of the different methods considered in this chapter the best solution found (**Sol**), the computational time in seconds (**Time**), the required number of CG iterations (**#Iters**) and the percentual deviation (**%Dev**) compared to the BKS are presented.

The results obtained through the use of GC+BBC and GC+CP seem to indicate that both methods outperform CG+ILP. It is observed that CPU time and quality of the solutions found are improved through the solution approaches. According to the results observed, both CG+BBC and CG+CP approaches might be used as efficient strategies to tackle the PS derived from the CG framework adopted to solve CMLP and CMLP-MR. The experiments indicate that CG+BBC and CG+CP find 67.5% and 93% of the optimal solutions for the evaluated instances respectively, while only 12.5% are returned by CG+ILP. Additionally, it is observed that all the instances solved by CG+BBC are also solved by CG+CP; it can be computed that the CPU time consumed for CG+BBC over the set of instances optimally solved is slightly lower than the required for CG+CP, *i.e.*, 277.3 seconds against 351.5 respectively. Regarding CG+BBC, the experiments indicate that the cause of a poor performance in the 33% of unsolved instances is to be sought in some particular iterations of CG. In those cases, the combined effect of the values for the dual variables and the inner characteristics of

Instance	m	n	R _s	Opt/BKS*	CG+ILP ¹				CG+BBC				CG+CP			
					Time	# Iters	Sol	% Dev	Time	# Iters	Sol	% Dev	Time	# Iters	Sol	% Dev
CMLP_MR_001	100	15	100	5.40	3600.0	75	5.00	7%	72.3	49	5.40	0%	3.2	86	5.40	0%
CMLP_MR_002	100	15	100	4.00	3600.0	54	3.61	10%	2.9	26	4.00	0%	2.4	54	4.00	0%
CMLP_MR_003	100	15	100	3.75	32.8	33	3.75	0%	11.4	39	3.75	0%	2.8	62	3.75	0%
CMLP_MR_004	100	15	100	4.00	140.2	67	4.00	0%	6.5	26	4.00	0%	6.4	65	4.00	0%
CMLP_MR_005	100	30	100	5.00	3600.0	192	4.61	8%	137.3	126	5.00	0%	11.5	108	5.00	0%
CMLP_MR_006	100	30	100	4.00	116.5	38	4.00	0%	21.2	31	4.00	0%	5.5	71	4.00	0%
CMLP_MR_007	100	30	100	3.00	3600.0	18	2.67	11%	27.9	18	3.00	0%	3.9	48	3.00	0%
CMLP_MR_008	100	30	100	4.00	72.8	38	4.00	0%	9.2	32	4.00	0%	5.1	70	4.00	0%
CMLP_MR_009	200	15	100	8.00	2943.6	273	8.00	0%	11.2	422	8.00	0%	29.1	126	8.00	0%
CMLP_MR_010	200	15	100	12.00	3600.0	354	10.03	16%	23.0	159	12.00	0%	49.5	207	12.00	0%
CMLP_MR_011	200	15	100	9.00	3600.0	321	8.87	1%	13.3	102	9.00	0%	21.4	149	9.00	0%
CMLP_MR_012	200	15	100	14.60	3600.0	289	9.58	34%	3600.0	144	11.96	18%	110.9	394	14.60	0%
CMLP_MR_013	200	30	100	8.00	3600.0	304	4.79	40%	13.8	211	8.00	0%	40.5	170	8.00	0%
CMLP_MR_014	200	30	100	7.00	3600.0	536	5.47	22%	4.6	118	7.00	0%	32.1	137	7.00	0%
CMLP_MR_015	200	30	100	9.00	3600.0	336	4.96	45%	14.8	151	9.00	0%	46.3	181	9.00	0%
CMLP_MR_016	200	30	100	11.40	3600.0	301	5.28	54%	1356.2	1015	11.40	0%	58.1	253	11.40	0%
CMLP_MR_017	300	15	100	15.00	3600.0	224	3.60	76%	51.6	117	15.00	0%	128.3	291	15.00	0%
CMLP_MR_018	300	15	100	21.25	3600.0	271	3.41	84%	3600.0	2313	19.59	8%	2585.1	1225	21.25	0%
CMLP_MR_019	300	15	100	12.00	3600.0	263	3.58	70%	25.0	150	12.00	0%	93.7	239	12.00	0%
CMLP_MR_020	300	15	100	16.00	3600.0	218	3.61	77%	250.9	378	16.00	0%	133.4	295	16.00	0%
CMLP_MR_021	300	30	100	15.00	3600.0	342	3.48	77%	209.2	290	15.00	0%	186.7	360	15.00	0%
CMLP_MR_022	300	30	100	13.20	3600.0	261	2.67	80%	110.2	214	13.20	0%	195.7	377	13.20	0%
CMLP_MR_023	300	30	100	12.00	3600.0	270	2.69	78%	42.6	292	12.00	0%	121.1	283	12.00	0%
CMLP_MR_024	300	30	100	13.00	3600.0	193	2.73	79%	61.1	208	13.00	0%	143.2	309	13.00	0%
CMLP_MR_025	400	15	100	18.60	3600.0	212	2.73	85%	3600.8	789	18.28	2%	447.0	406	18.60	0%
CMLP_MR_026	400	15	100	18.00	3600.0	147	2.65	85%	101.2	472	18.00	0%	470.4	421	18.00	0%
CMLP_MR_027	400	15	100	22.00	3600.0	179	2.66	88%	33.7	259	22.00	0%	541.3	459	22.00	0%
CMLP_MR_028	400	15	100	17.00	3600.0	230	2.68	84%	40.9	350	17.00	0%	387.3	371	17.00	0%
CMLP_MR_029	400	30	100	17.60	3600.0	253	2.58	85%	181.6	391	17.60	0%	499.2	429	17.60	0%
CMLP_MR_030	400	30	100	18.00	3600.0	188	2.31	87%	63.3	339	18.00	0%	576.3	475	18.00	0%
CMLP_MR_031	400	30	100	21.00	3600.0	153	2.35	89%	56.3	509	21.00	0%	750.3	550	21.00	0%
CMLP_MR_032	400	30	100	17.00	3600.0	120	2.39	86%	78.7	536	17.00	0%	538.0	450	17.00	0%
CMLP_MR_033	500	15	100	30.00	3600.0	125	2.23	93%	3600.0	3307	24.31	19%	1855.2	690	30.00	0%
CMLP_MR_034	500	15	100	27.00	3600.0	146	2.24	92%	385.5	674	27.00	0%	1667.6	647	27.00	0%
CMLP_MR_035	500	15	100	31.00	3600.0	121	2.17	93%	414.2	1041	31.00	0%	1700.7	657	31.00	0%
CMLP_MR_036	500	15	100	28.00	3600.0	206	2.37	92%	235.3	840	28.00	0%	1482.3	607	28.00	0%
CMLP_MR_037	500	30	100	23.00	3600.0	102	3.61	87%	317.7	752	23.00	0%	1490.4	608	23.00	0%
CMLP_MR_038	500	30	100	27.00	3600.0	123	5.03	81%	603.3	772	27.00	0%	1798.9	672	27.00	0%
CMLP_MR_039	500	30	100	27.00	3600.0	75	6.53	77%	225.7	860	27.00	0%	1803.8	678	27.00	0%
CMLP_MR_040	500	30	100	28.00	3600.0	156	3.87	86%	1537.1	1699	28.00	0%	1915.1	704	28.00	0%
CMLP_MR_041	100	15	125	6.40	3600.0	87	6.01	6%	1785.5	172	6.40	0%	4.7	107	6.40	0%
CMLP_MR_042	100	15	125	4.40	34.7	39	4.40	0%	17.9	109	4.40	0%	2.8	58	4.40	0%
CMLP_MR_043	100	15	125	3.75	16.9	25	3.75	0%	14.1	67	3.75	0%	2.6	52	3.75	0%
CMLP_MR_044	100	15	125	6.56	3600.0	38	5.83	11%	3600.0	371	5.93	10%	7.7	168	6.56	0%
CMLP_MR_045	100	30	125	6.40	3600.0	123	5.39	16%	3600.0	528	6.20	3%	9.3	194	6.40	0%
CMLP_MR_046	100	30	125	4.40	58.4	41	4.40	0%	3.0	77	4.40	0%	3.1	63	4.40	0%
CMLP_MR_047	100	30	125	3.75	50.9	33	3.75	0%	36.3	124	3.75	0%	2.6	51	3.75	0%
CMLP_MR_048	100	30	125	6.40	3600.0	247	5.59	13%	3600.0	723	6.32	1%	11.0	215	6.40	0%
CMLP_MR_049	200	15	125	9.60	3600.0	323	8.45	12%	36.0	188	9.60	0%	19.8	141	9.60	0%
CMLP_MR_050	200	15	125	16.80	3600.0	319	8.23	51%	3600.0	618	14.31	15%	69.6	378	16.80	0%
CMLP_MR_051	200	15	125	16.00	3600.0	353	9.21	42%	194.6	447	16.00	0%	169.7	553	16.00	0%
CMLP_MR_052	200	15	125	15.00	3600.0	331	8.88	41%	105.7	277	15.00	0%	49.0	270	15.00	0%
CMLP_MR_053	200	30	125	9.60	3600.0	460	9.09	5%	86.9	1034	9.60	0%	25.2	167	9.60	0%
CMLP_MR_054	200	30	125	11.00	3600.0	280	3.48	68%	46.2	1086	11.00	0%	35.8	213	11.00	0%
CMLP_MR_055	200	30	125	15.00	3600.0	287	3.57	76%	3600.0	1528	14.44	4%	152.8	518	15.00	0%
CMLP_MR_056	200	30	125	14.80	3600.0	350	3.57	76%	3600.0	1137	13.89	6%	69.2	331	14.80	0%
CMLP_MR_057	300	15	125	18.40	3600.0	283	3.61	80%	3600.0	1640	16.81	9%	147.3	312	18.40	0%
CMLP_MR_058	300	15	125	21.25	3600.0	234	3.66	83%	3600.0	973	19.91	6%	277.2	455	21.25	0%
CMLP_MR_059	300	15	125	15.20	3600.0	277	3.53	77%	143.7	343	15.20	0%	125.2	291	15.20	0%
CMLP_MR_060	300	15	125	19.00	3600.0	238	3.56	81%	3600.0	413	17.76	7%	168.4	339	19.00	0%
CMLP_MR_061	300	30	125	18.40	3600.0	383	3.53	81%	3600.0	627	17.87	3%	182.0	350	18.40	0%
CMLP_MR_062	300	30	125	18.40	3600.0	201	2.54	86%	3600.0	1367	15.22	17%	260.1	436	18.40	0%
CMLP_MR_063	300	30	125	15.20	3600.0	156	2.54	83%	841.5	1720	15.20	0%	146.3	315	15.20	0%
CMLP_MR_064	300	30	125	14.60	3600.0	146	2.62	82%	1116.6	3578	14.60	0%	129.8	286	14.60	0%
CMLP_MR_065	400	15	125	20.00	3600.0	228	2.67	87%	195.5	552	20.00	0%	494.7	430	20.00	0%
CMLP_MR_066	400	15	125	26.00	3600.0	30	1.00	96%	3600.0	937	23.37	10%	849.6	594	26.00	0%
CMLP_MR_067	400	15	125	30.00	3600.0	139	2.47	92%	2496.3	3567	30.00	0%	658.3	510	30.00	0%
CMLP_MR_068	400	15	125	29.00	3600.0	193	2.52	91%	3600.0	1323	25.47	12%	751.2	548	29.00	0%
CMLP_MR_069	400	30	125	19.60	3600.0	205	2.26	88%	3600.0	977	18.89	4%	560.4	458	19.60	0%
CMLP_MR_070	400	30	125	26.00	3600.0	105	2.29	91%	3600.0	875	22.73	13%	897.2	608	26.00	0%
CMLP_MR_071	400	30	125	25.40	3600.0	123	1.87	93%	3600.0	1824	23.12	9%	701.2	528	25.40	0%
CMLP_MR_072	400	30	125	23.20	3600.0	141	2.06	91%	3600.0	750	22.28	4%	677.2	513	23.20	0%
CMLP_MR_073	500	15	125	32.60	3600.0	114	2.27	93%	3600.0	1001	26.39	19%	1745.6	664	32.60	0%
CMLP_MR_074	500	15	125	31.20	3600.0	128	2.24	93%	1930.2	1075	31.20	0%	1643.3	641	31.20	0%
CMLP_MR_075	500	15	125	49.00	3600.0	111	2.32	95%	3600.0	748	37.77	23%	3600.0	993	44.55	9%
CMLP_MR_076	500	15	125	41.00	3600.0	123	2.05	95%	3600.0	1327	37.81	8%	3600.0	995	40.35	2%
CMLP_MR_077	500	30	125	32.40	3600.0	138	2.32	93%	3600.0	2407	28.22	13%	2890.3	884	32.40	0%
CMLP_MR_078	500	30	125	31.20	3600.0	105	1.00	97%	3600.0	132	28.64	8%	1929.0	699	31.20	0%
CMLP_MR_079	500	30	125	40.00	3600.0	312	1.00	98%	3600.0	727	35.64	11%	3600.0	988	39.28	2%
CMLP_MR_080	500	30	125	41.00	3600.0	501	2.00	95%	3600.0							

Instance	m	n	R_s	Opt/BKS*	CG+ILP ²				CG+BBC				CG+CP			
					Time	# Iters	Sol	% Dev	Time	# Iters	Sol	%	Time	# Iters	Sol	% Dev
CMLP_MR_081	100	15	100	5.00	843.2	227	5.00	0%	56.8	96	5.00	0%	3.1	80	5.00	0%
CMLP_MR_082	100	15	100	4.00	229.7	216	4.00	0%	13.6	60	4.00	0%	2.4	59	4.00	0%
CMLP_MR_083	100	15	100	3.00	108.1	125	3.00	0%	52.0	75	3.00	0%	2.0	44	3.00	0%
CMLP_MR_084	100	15	100	4.00	168.8	184	4.00	0%	3.2	27	4.00	0%	2.8	71	4.00	0%
CMLP_MR_085	100	30	100	5.00	3600.0	273	3.50	30%	712.0	331	5.00	0%	5.4	134	5.00	0%
CMLP_MR_086	100	30	100	4.00	3600.0	25	1.01	75%	7.2	33	4.00	0%	3.1	78	4.00	0%
CMLP_MR_087	100	30	100	3.00	80.6	130	3.00	0%	13.2	38	3.00	0%	2.4	52	3.00	0%
CMLP_MR_088	100	30	100	4.00	3600.0	278	3.66	9%	29.9	32	4.00	0%	3.4	83	4.00	0%
CMLP_MR_089	200	15	100	8.00	351.9	65	8.00	0%	11.7	107	8.00	0%	18.8	147	8.00	0%
CMLP_MR_090	200	15	100	12.00	3600.0	163	9.85	18%	43.7	386	12.00	0%	46.8	274	12.00	0%
CMLP_MR_091	200	15	100	9.00	3600.0	1051	8.80	2%	13.3	139	9.00	0%	24.4	178	9.00	0%
CMLP_MR_092	200	15	100	12.00	3600.0	1050	9.12	24%	137.9	308	12.00	0%	36.4	235	12.00	0%
CMLP_MR_093	200	30	100	8.00	3600.0	379	7.26	9%	26.7	140	8.00	0%	22.9	168	8.00	0%
CMLP_MR_094	200	30	100	7.00	3600.0	1151	4.41	37%	4.9	82	7.00	0%	21.9	159	7.00	0%
CMLP_MR_095	200	30	100	9.00	3600.0	1038	6.80	24%	25.1	153	9.00	0%	28.8	198	9.00	0%
CMLP_MR_096	200	30	100	11.00	3600.0	964	7.92	28%	162.0	479	11.00	0%	37.1	235	11.00	0%
CMLP_MR_097	300	15	100	15.00	3600.0	644	3.10	79%	101.6	224	15.00	0%	142.6	322	15.00	0%
CMLP_MR_098	300	15	100	17.00	3600.0	559	2.59	85%	1753.4	753	17.00	0%	258.0	455	17.00	0%
CMLP_MR_099	300	15	100	12.00	3600.0	581	2.81	77%	30.7	158	12.00	0%	104.4	268	12.00	0%
CMLP_MR_100	300	15	100	16.00	3600.0	666	2.89	82%	316.6	1096	16.00	0%	185.6	376	16.00	0%
CMLP_MR_101	300	30	100	15.00	3600.0	614	2.94	80%	209.2	612	15.00	0%	219.0	410	15.00	0%
CMLP_MR_102	300	30	100	13.00	3600.0	659	2.88	78%	110.8	443	13.00	0%	186.8	379	13.00	0%
CMLP_MR_103	300	30	100	12.00	3600.0	700	3.35	72%	64.1	386	12.00	0%	117.5	289	12.00	0%
CMLP_MR_104	300	30	100	13.00	3600.0	660	2.94	77%	61.5	437	13.00	0%	153.8	337	13.00	0%
CMLP_MR_105	400	15	100	16.00	3600.0	467	2.29	86%	343.6	1595	16.00	0%	364.8	361	16.00	0%
CMLP_MR_106	400	15	100	18.00	3600.0	496	2.45	86%	111.1	722	18.00	0%	567.8	458	18.00	0%
CMLP_MR_107	400	15	100	22.00	3600.0	447	2.44	89%	98.0	1423	22.00	0%	957.9	507	22.00	0%
CMLP_MR_108	400	15	100	17.00	3600.0	473	2.55	85%	126.7	628	17.00	0%	586.9	373	17.00	0%
CMLP_MR_109	400	30	100	16.00	3600.0	475	2.39	85%	181.7	788	16.00	0%	692.3	412	16.00	0%
CMLP_MR_110	400	30	100	18.00	3600.0	502	2.49	86%	199.6	781	18.00	0%	942.4	511	18.00	0%
CMLP_MR_111	400	30	100	21.00	3600.0	493	2.44	88%	94.7	1202	21.00	0%	1162.0	560	21.00	0%
CMLP_MR_112	400	30	100	17.00	3600.0	479	2.36	86%	101.9	759	17.00	0%	863.9	492	17.00	0%
CMLP_MR_113	500	15	100	30.00	3600.0	378	2.05	93%	3600.0	3307	24.31	19%	3205.6	788	30.00	0%
CMLP_MR_114	500	15	100	27.00	3600.0	381	2.07	92%	385.6	1398	27.00	0%	2266.1	735	27.00	0%
CMLP_MR_115	500	15	100	31.00	3600.0	357	2.26	93%	414.3	2367	31.00	0%	2091.8	762	31.00	0%
CMLP_MR_116	500	15	100	28.00	3600.0	368	2.00	93%	235.7	1713	28.00	0%	1714.4	675	28.00	0%
CMLP_MR_117	500	30	100	23.00	3600.0	383	2.00	91%	317.2	992	23.00	0%	1501.8	627	23.00	0%
CMLP_MR_118	500	30	100	27.00	3600.0	368	2.00	93%	2151.2	3722	27.00	0%	2279.4	751	27.00	0%
CMLP_MR_119	500	30	100	27.00	3600.0	344	2.21	92%	1133.6	2758	27.00	0%	2207.4	707	27.00	0%
CMLP_MR_120	500	30	100	28.00	3600.0	323	2.15	92%	3600.0	4606	24.36	13%	2499.5	756	28.00	0%
CMLP_MR_121	100	15	125	6.00	3600.0	67	4.86	19%	132.8	90	6.00	0%	4.4	112	6.00	0%
CMLP_MR_122	100	15	125	4.00	57.2	95	4.00	0%	5.0	35	4.00	0%	2.3	55	4.00	0%
CMLP_MR_123	100	15	125	3.00	20.0	51	3.00	0%	7.8	32	3.00	0%	1.8	38	3.00	0%
CMLP_MR_124	100	15	125	6.00	3600.0	181	4.63	23%	240.5	164	6.00	0%	6.8	162	6.00	0%
CMLP_MR_125	100	30	125	6.00	3600.0	92	5.28	12%	186.7	120	6.00	0%	9.6	213	6.00	0%
CMLP_MR_126	100	30	125	4.00	118.0	113	4.00	0%	3.3	31	4.00	0%	2.4	57	4.00	0%
CMLP_MR_127	100	30	125	3.00	59.0	68	3.00	0%	18.7	34	3.00	0%	2.1	42	3.00	0%
CMLP_MR_128	100	30	125	6.00	3600.0	146	4.26	29%	3600.0	84	5.37	11%	120.5	551	6.00	0%
CMLP_MR_129	200	15	125	9.00	737.0	348	9.00	0%	21.2	472	9.00	0%	18.1	141	9.00	0%
CMLP_MR_130	200	15	125	16.00	3600.0	375	10.37	35%	2263.6	696	16.00	0%	60.2	316	16.00	0%
CMLP_MR_131	200	15	125	15.00	3600.0	354	9.34	38%	3600.0	783	14.03	6%	202.1	614	15.00	0%
CMLP_MR_132	200	15	125	12.00	3600.0	258	9.46	21%	194.3	256	12.00	0%	28.8	197	12.00	0%
CMLP_MR_133	200	30	125	9.00	3600.0	433	8.51	5%	5.7	360	9.00	0%	22.3	163	9.00	0%
CMLP_MR_134	200	30	125	11.00	3600.0	435	8.44	23%	10.6	963	11.00	0%	33.6	213	11.00	0%
CMLP_MR_135	200	30	125	15.00	3600.0	393	8.56	43%	3600.0	761	13.37	11%	3600.0	1249	14.72	2%
CMLP_MR_136	200	30	125	12.00	3600.0	3987	8.69	28%	3600.0	372	11.35	5%	34.2	218	12.00	0%
CMLP_MR_137	300	15	125	18.00	3600.0	793	3.90	78%	3600.0	1764	16.83	6%	152.9	333	18.00	0%
CMLP_MR_138	300	15	125	17.00	3600.0	604	4.76	72%	3600.0	1164	14.13	17%	185.1	366	17.00	0%
CMLP_MR_139	300	15	125	15.00	3600.0	772	4.25	72%	197.3	359	15.00	0%	200.0	309	15.00	0%
CMLP_MR_140	300	15	125	17.00	3600.0	1107	4.20	75%	3600.0	423	15.67	8%	218.7	329	17.00	0%
CMLP_MR_141	300	30	125	18.00	3600.0	915	4.09	77%	3600.0	464	15.67	13%	248.7	359	18.00	0%
CMLP_MR_142	300	30	125	17.00	3600.0	804	3.59	79%	3600.0	336	15.14	11%	332.0	437	17.00	0%
CMLP_MR_143	300	30	125	15.00	3600.0	1269	3.76	75%	32.5	472	15.00	0%	211.7	343	15.00	0%
CMLP_MR_144	300	30	125	14.00	3600.0	1187	3.74	73%	74.0	744	14.00	0%	158.2	305	14.00	0%
CMLP_MR_145	400	15	125	16.00	3600.0	931	2.82	82%	16.7	816	16.00	0%	329.8	322	16.00	0%
CMLP_MR_146	400	15	125	25.50	3600.0	733	3.23	87%	3600.0	688	21.34	16%	1387.9	722	25.50	0%
CMLP_MR_147	400	15	125	29.00	3600.0	704	3.04	90%	3600.0	3096	28.77	1%	700.4	508	29.00	0%
CMLP_MR_148	400	15	125	29.00	3600.0	546	2.67	91%	3600.0	3780	24.66	15%	1096.5	688	29.00	0%
CMLP_MR_149	400	30	125	16.00	3600.0	503	3.10	81%	11.1	1164	16.00	0%	354.1	329	16.00	0%
CMLP_MR_150	400	30	125	25.50	3600.0	550	3.48	86%	3600.0	1290	23.43	8%	1453.4	745	25.50	0%
CMLP_MR_151	400	30	125	25.00	3600.0	572	3.00	88%	122.5	768	25.00	0%	897.8	568	25.00	0%
CMLP_MR_152	400	30	125	23.00	3600.0	757	3.21	86%	3600.0	2576	22.51	2%	920.4	574	23.00	0%
CMLP_MR_153	500	15	125	30.00	3600.0	618	2.72	91%	3600.0	1791	26.14	13%	1824.6	648	30.00	0%
CMLP_MR_154	500	15	125	31.00	3600.0	529	3.57	88%	3600.0	834	28.85	7%	2335.2	719	31.00	0%
CMLP_MR_155	500	15	125	45.06*	3600.0	623	2.99	93%	3600.0	580	35.26	22%	3600.0	913	42.20	6%
CMLP_MR_156	500	15	125	41.00	3600.0	560	2.53	94%	3600.0	4216	32.99	20%	3600.0	1012	37.35	9%
CMLP_MR_157	500	30	125	30.00	3600.0	662	3.05	90%	3600.0	1773	26.30	12%	2237.7	711	30.00	0%
CMLP_MR_158	500	30	125	31.00	3600.0	468	2.40	92%	3600.0	1590	28.88	7%	2655.3	777	31.00	0%
CMLP_MR_159	500	30	125	40.00	3600.0	593	3.04	92%	3600.0	1009	34.60	14%	3600.0	1085	36.75	8%
CMLP_MR_160	500	30	125	37.83*	3600.0	549	3.20	92%								

the instance leads to longer computational times. Hence, the solution process may get stuck without any improvement until time limit is reached. In the case of CG+CP it may be partially explained by a slower convergence of CG that leads to an increased number of iterations. As a consequence, the method might be unable to find the optimal solution within the time limit.

As previously mentioned, by limiting the number of active sensors in a partition it is possible to decrease the number of CG iterations necessary to find an optimal solution, nonetheless it might come at expenses of an increase on the average CPU time required per iteration. Figures 5.3a and 5.3b depict this phenomenon and present the effect of the acceleration procedure introduced in Section 4.1 combined with the BBC strategy. As observed, the classical implementation of CG+BBC is prone to present some plateaus at the beginning of the optimization process that ultimately lead to useless CG iterations and an increase of the CPU time and the number of iterations required to achieve the optimal solution. As observed, the proposed approach is able to reduce the required number of iterations of CG, a problem that is especially evident when using CG+BBC.

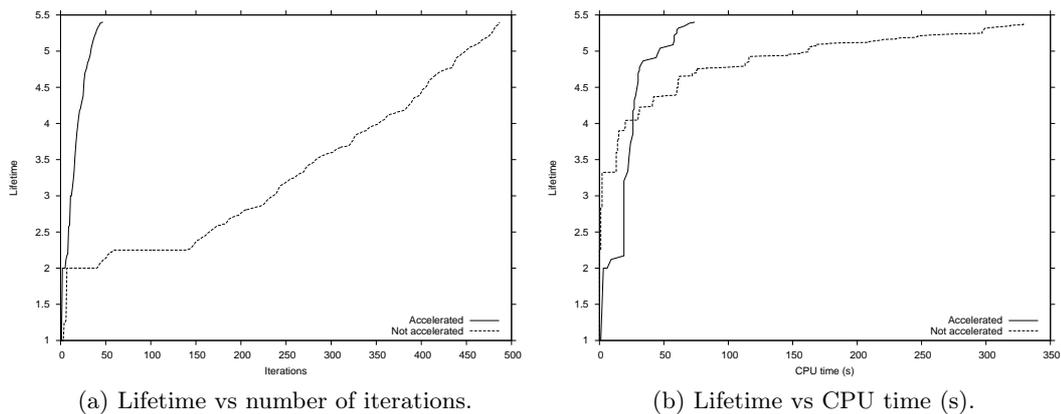


Figure 5.3 – Comparison of objective function evolution throughout CG for instance CMLP_MR_001.

According to the experimental results, the performance CG+BBC might be affected by the characteristics of the instances. As it might be expected, the CPU time required to solve the problems is related with the number of available nodes in the network. However, the impact of the increase in the sensing range seems to be bigger and leads to higher computational times, probably as an effect of the increment in the density of the graphs or the availability of sensing links to be established that may increase the computational effort required to solve PS. On the other hand, the performance of CG+CP is virtually neither affected by characteristics as the sensing or communication range, nor for the characteristics of the energy consumption in sensors (bimodal vs. multimodal). Nonetheless, an expected increase in CPU time is observed as the size of the instances gets bigger.

5.6 Conclusions

This chapter explores the use of exact approaches in order to maximize the lifetime in wireless sensor networks operating under coverage and connectivity constraints. A CG algorithm exploiting separately a Branch-and-Cut algorithm based on Benders' decomposition and a constraint programming approach is proposed. The former is used to take advantage of the structure of the network design problem arising in PS. The latter is levered by dual cost profile to focus on propagation on the network.

An extensive set of experiments demonstrates that both algorithms outperform the results obtained through the single utilization of an ILP solver throughout iterations of the CG algorithm. The results indicate that the BBC and CP approaches can be used efficiently to tackle the pricing subproblem and to reduce the computational time required to solve CMLP-MR and CMLP. Moreover, these results are promising as they might indicate that similar approaches based on CG can be used to maximize network lifetime in WSN containing more sophisticated sensors with more capabilities than in this chapter. Moreover, it indicates that CG can be used as an efficient method to guarantee the optimality of the solutions found and to provide practitioners, maybe considering distributed approaches, with an optimal upper bound to evaluate their proposals.

Further research will consider specific features about strategies to improve the performance of both methods when solving the pricing subproblem and its extension to network design problems and energy usage optimization in wireless networks. Hybrid approaches combining the proposed exact methods with (meta)heuristic approaches in order to improve the efficiency of the proposed CG methods will be considered as well. Finally, the extension of the method to consider variants of the problem in which partial coverage of the targets is allowed (*e.g.*, quality of service) and more sophisticated energy consumption models for the sensors, *e.g.*, considering the cost of establishing a connection between nodes and targets, is a promising future research direction.



Partial coverage to extend the lifetime in wireless multi-role sensor networks

This chapter studies the effect of partial coverage to extend the lifetime of wireless sensor networks and introduces a hybrid exact solution strategy that efficiently maximizes network lifetime. We consider a set of sensors used to provide coverage of discrete targets and transfer the information to the sink node via multi-hop wireless communication. An active sensor has one of the two roles: it is a source when it senses and transfer data; it is a relay if it only transfers data. Network lifetime is extended through the use of covers representing sensors role allocation and network topology. A hybrid column generation is used to identify the optimal operation schedules and roles allocation that maximize network lifetime. A constraint programming strategy is provided to identify profitable network configurations leading to extend network lifetime and to prove optimality of the solutions found. Moreover, an evolutionary algorithm is proposed to boost up the solution process and accelerate convergence. Extensive computational results demonstrate the effectiveness of the proposed approach.

6.1 Introduction

Recent advances in microelectronic technologies have levered the capabilities offered by wireless sensors and extended their adoption. These newly introduced characteristics have allowed to exploit wireless sensors networks (WSN), made of hundreds to thousands sensors, in many application domains [Biagioni and Bridges, 2002, Chen et al., 2013, Kim et al., 2007, Werner-Allen et al., 2006]. Wireless sensors networks have been successfully employed in applications where the use of traditional wired sensor networks was neither practical nor feasible [Gaura et al., 2010b]. WSN can be used to monitor remote, hostile or inaccessible areas through the collaboration of individual sensors that store the information collected or transmit it to the final user via multi-hop wireless communications [Keskin et al., 2014].

Wireless sensors are typically battery powered devices with limited computation, communication and memory resources [Yu et al., 2009]. Battery limitations impose a major concern regarding the application of WSN to realistic scenarios. If an interesting phenomenon is scarcely covered by the sensor nodes, the time during which network can provide coverage is limited by those sensors. In the same way, if connectivity to a sink (to

collect information) is required, it is expected that sensors surrounding it will deplete their energy at a fastest pace since they carry all the data collected by the sensors. As a consequence, the isolation of the sink can appear due to the power exhaustion of neighboring sensors (*the Hotspot problem* [Abdulla et al., 2012]) limiting network lifetime. These phenomena stress the need for an efficient use of energy to control the workload associated to each sensor with the objective of guaranteeing coverage while extending network lifetime.

In some applications is not necessary to fully cover the interesting area or targets in order to have a general idea of the system state [Li et al., 2011]. In that case, partial coverage is beneficial to WSN lifetime [Chen et al., 2010, Wang and Kulkarni, 2006]. Raiconi and Gentili [Gentili and Raiconi, 2013] evaluate the effect of partial coverage in wireless sensor networks in which connectivity is not required and demonstrate the benefits in terms of network lifetime. Similarly, the results presented by Castaño *et al.* [Castaño et al., 2014] demonstrate that, in connectivity constrained WSN, partially covering the set of targets can significantly extend network lifetime especially when lifetime is not bounded by the hotspot problem.

Current technology allows sensors to adopt different energy consumption profiles upon the basis of the tasks they perform within the network. The use of sensors consuming energy at different rates has been successfully explored in the context of sensors having adjustable sensing ranges demonstrating the effectiveness to extend networks lifetime [Cardei et al., 2006, Cerulli et al., 2012, Dhawan et al., 2006, Lu et al., 2005, Rossi et al., 2012b]. The present research considers a different approach in which connectivity is enforced and sensors can adopt up to three different energy consumption profiles [Zhao and Gurusamy, 2008]. Sensors might be used to monitor the interesting phenomenon (or variables) and transmit the information to other sensors or to the sink through multi-hop wireless communication. Moreover, and considering that redundant coverage can lead to energy waste, sensors might be used only for transmission supporting other sensors that cannot reach the sink to redirect the collected information to this one. Finally, sensors that are not necessary neither for sensing nor for transmission can adopt an idle state in which the energy consumption is minimal.

Column generation has demonstrated to be an efficient strategy to solve coverage and scheduling problems in WSN [Castaño et al., 2014, Gu et al., 2009c, Raiconi and Gentili, 2011, Rossi et al., 2012b]. In this chapter column generation (CG) is adopted to maximize the lifetime in connectivity constrained WSN used to provide coverage of discrete targets. Previous approaches based on CG to solve WSN problems in which partial coverage of those targets is allowed are extended to consider sensors that can adopt different energy consumption profiles according to the tasks they perform. In order to efficiently address the pricing subproblem (PS) and identify interesting cover sets, a constraint programming based approach, and an evolutionary algorithm to solve PS, and accelerate the convergence of CG, are proposed. Finally, it is demonstrated that by hybridizing both approaches it is possible to improve the performance of the method with the objective of obtaining optimal solutions.

This chapter is structured as follows: Section 6.2 introduces the maximum network lifetime problem in wireless multi-role sensor networks with coverage and connectivity

constraints and propose a mathematical model for it. In Section 6.3 the solution approach based on CG is introduced as well as the detailed description of the approaches adopted to efficiently use CG to solve the problem. The experimental results and the performance evaluation of the method is described in Section 6.4. Finally, conclusions and future directions of research are proposed in Section 6.5.

6.2 Problem description and model

This section introduces the maximum network lifetime problem in WSN with coverage and connectivity constraints and presents an overview of the algorithmic approaches proposed to address related problems. Then, a mathematical description of the problem is proposed and the characteristics of the problem are highlighted.

In densely deployed WSN, where more sensors than required to provide coverage are available, it is possible to extend lifetime by activating sequentially subsets of sensors. The general idea is to put a set of sensors in active mode (cover sets) and to schedule subsets of active sensors over time while ensuring that coverage and connectivity to the sink is respected at any moment [Cardei et al., 2005, Castaño et al., 2013, 2014, Lu et al., 2005, Zorbas et al., 2010]. The same idea can be extended to WSN in which sensors can adopt different consumption profiles by allocating them tasks within the network throughout network lifetime at different time intervals guaranteeing that battery lifetime of each sensor node is respected.

According to the nature of the covers used, two main approaches to extend network lifetime in WSN can be devised: disjoint [Jia et al., 2008, Slijepcevic and Potkonjak, 2002] and non-disjoint [Cardei and Cardei, 2008b, Cardei et al., 2005, Castaño et al., 2013, 2014]. Although both pursue the same objective, typically they differ in the way they are applied and the benefits and disadvantages offered by each. In the one hand, by using disjoint subsets the objective is typically to maximize the number of covers. Nonetheless, this can lead to shortest lifetimes and inefficient use of sensors energy; in contrast, it offers robustness in the sense that each sensor belongs to at most one subset and failures only affect the quality of coverage of a single set [Henna and Erlebach, 2013]. In the second hand, non-disjoint subsets lead to longer lifetimes but lacks in fault tolerance because a sensor that fails may participate in more than one cover [Henna and Erlebach, 2013, Wang et al., 2009]. Nonetheless, the benefits obtained by using non-disjoint subsets can be enough to permit the lack of some sensor nodes.

Non-disjoint approaches have been largely considered as strategies to maximize the lifetime in WSN [Liu et al., 2011, 2007]. Yu Gu *et al.* [Gu et al., 2009c] propose a solution method based on column generation to address a lifetime problem in WSN and then demonstrate that, in general, the schedules of sensor nodes in target coverage problems can be represented by using patterns (indicating the energy consumption rate of each sensor) [Gu et al., 2011]. This latter conclusion lead to presume that CG is an efficient strategy, and is flexible enough to consider different characteristics of the network operation.

In this chapter a non-disjoint global approach to maximize the lifetime in WSN is

adopted. The network is characterized by a set of sensors $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$ randomly deployed to monitor a set of targets $\mathcal{K} = \{k_1, k_2, \dots, k_n\}$ and to transfer sensing information to the sink, denoted by S_0 . Each sensor can operate at three different energy consumption profiles according to the task it performs in the network. A sensor can be used to monitor the targets located within its sensing range R_s and transmit the collected information directly to the sink or to other sensors located within its communication range R_c , in this case the sensor is called a *source* and consumes energy at a rate E_s . In addition to its own collected data, a source may also transfer information sent by other sensors. A sensor might be used only to retransmit the information collected by other sensors when they are not able to establish direct communication with the sink. This situation is specially desirable when lifetime is limited by sensors neighboring the sink, in this case the sensors are called *relays* and consume energy at a rate E_r . Finally, a sensor that is neither used for sensing nor for transmitting is called *idle* and consumes energy at a lower energy consumption rate E_i . Typically $E_s > E_r > E_i$, however this assumption is not necessary in our approach.

A graph $G(\mathcal{K} \cup \mathcal{S} \cup \{S_0\}, \mathbb{E})$ is used to represent all the possible connections sensor-sensor, sensor-target and sensor-sink in the network. The set \mathbb{E} is a set of edges where $e(u, v) \in \mathbb{E}$ exists if: (i) $u \in \mathcal{K}$, $v \in \mathcal{S}$ and u is located within the sensing range R_s of v , (ii) $u, v \in \mathcal{S}$ and they are located within the communication range of each other or (iii) if the sink S_0 is located within the communication range of a sensor $u \in \mathcal{S}$. A partition P_j divides the set \mathcal{S} into three non-overlapping sets \mathcal{S}_s^j (source nodes), \mathcal{S}_r^j (relay nodes) and \mathcal{S}_i^j (idle nodes). $K^j \subseteq \mathcal{K}$ identifies a subset of targets to be monitored by the sensors belonging to \mathcal{S}_s^j .

Let $\mathcal{N}_j = \{S_0\} \cup K^j \cup \mathcal{S} \setminus \mathcal{S}_i^j$ be a set of nodes. A partition P_j is called valid if it induces a tree $G' = G[\mathcal{N}_j]$ where G' is the subgraph of G induced by \mathcal{N}_j . G' is such that there exists a path to send the information obtained from each target $k_j \in K^j$ to the sink, where $|K^j| \geq \lceil \alpha \mathcal{K} \rceil$. The values $\alpha \in [0, 1]$ denote the fraction of targets that must be covered by a valid partition and $\alpha < 1$ indicates that partial coverage of the targets is allowed, *i.e.*, it is not necessary to monitor the full set of targets.

In Figure 6.1a a wireless sensor network deployed to monitor 5 targets is illustrated. The dotted lines indicate all the possible *communication links* that can be established between sensors or the *coverage links* that can be established between targets and sensors. Partitions $P_j = \{\mathcal{S}_i^j, \mathcal{S}_r^j, \mathcal{S}_s^j\}$ of the wireless sensor network are represented in Figures 6.1b, 6.1c and 6.1d. Figures 6.1b and 6.1c are equivalent and represent two alternative connectivity trees implicit in the partition $P_j = \{\{s_1, s_2, s_4, s_5, s_7\}, \{s_8\}, \{s_0, s_3, s_6\}\}$. As observed, the target κ_5 is not covered; consequently, the fraction of coverage achieved is $\alpha = 0.8$, *i.e.*, 80% of target coverage. Figure 6.1d represents one subtree induced by the partition $P_j = \{\{s_0, s_3, s_4\}, \{s_2, s_5, s_8\}, \{s_1, s_6, s_7\}\}$ and reaches 100% of coverage. As observed, the encoding of the tree structure is not necessary as it can be easily derived from the structure of the partition and the only important aspect is to guarantee the existence of such a tree rooted at the sink and spanning all or a subset of the target nodes.

Let Ω denotes the set of all feasible partitions P_j , the maximum network lifetime

problem with role allocation, partial coverage and connectivity constraints (α -CMLP-MR) is to allocate the optimal time t_j for these partitions to be used; so as to maximize lifetime while respecting the battery's capacity b_{s_u} of the sensors. In other words, it consists in finding the optimal pairs (P_j, t_j) that maximize lifetime and guarantees a minimum level of coverage during network lifetime.

The binary parameter $y_{s_u \ell j}$ indicate that sensor s_u assumes the energy consumption rate $\ell \in \{E_i, E_r, E_s\}$ in the partition P_j . Assuming that the set Ω is completely known, as well as the values $y_{s_u \ell j}$, it is possible to formulate the problem by using the following linear programming model (RMP):

Model RMP : α -CMLP-MR

$$\text{Maximize: } \sum_{j|P_j \in \Omega} t_j \quad (6.1)$$

$$\sum_{j|P_j \in \Omega} (E_s y_{s_u E_s j} + E_r y_{s_u E_r j} + E_i y_{s_u E_i j}) t_j \leq b_{s_u} \quad \forall s_u \in \mathcal{S} \quad (6.2)$$

$$t_j \geq 0 \quad \forall j|P_j \in \Omega \quad (6.3)$$

The objective of model (6.1-6.3) is to maximize network lifetime (Eq. 6.1) by using the partitions $P_j \in \Omega$. Constraints (Eq. 6.2) are used to guarantee that the initial amount of energy b_{s_u} of each sensor is not exceeded. As expected, the time variables t_j , associated to the partitions, are continuous (Eq. 6.3).

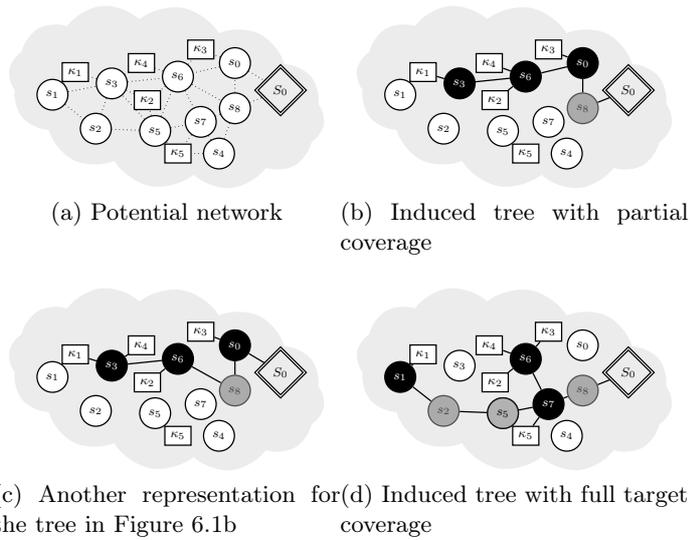


Figure 6.1 – A wireless sensor network providing full and partial coverage

6.3 Solution approach

As the number of valid partitions that can be obtained from Ω grows exponentially with the number of sensors in the network, the model (6.1-6.3) is defined over an exponential set of variables, and it might be impracticable to fully enumerate the set $P_j \in \Omega$. Moreover, even if they were enumerated, it would be observed that, in the optimal solution, only a few partitions are used for a non-zero time. As a consequence, a strategy leading to generate only the partitions useful to extend network lifetime can be efficiently adapted to solve α -CMLP-MR.

Column generation has been successfully applied to solve coverage and scheduling problems in WSN [Castaño et al., 2013, 2014, Gentili and Raiconi, 2013, Gu et al., 2009b, Raiconi and Gentili, 2011, Singh et al., 2013]. Previous research indicates that these approaches are “flexible” and can consider different characteristics of the WSN operation. By using this strategy, it is possible to maximize network lifetime by splitting the problem into two: (i) A restricted master problem (RMP) and (ii) A pricing subproblem (PS). The former is used to allocate the optimal time interval that valid partitions have to be used. The latter is used to identify new interesting partitions, or allocation of roles to the sensors, that satisfy the coverage and connectivity requirements and might be used to further extend lifetime.

In the CG method, an initial solution is provided, *i.e.*, an initial set of interesting partitions, and is used as the basis to further improve network lifetime. In other words, CG starts with a RMP based on a subset of partitions $\Omega' \subseteq \Omega$. Then, based on the optimal solution for this RMP, it is possible to compute new partitions, columns for RMP, based on the reduced cost criterion. The reduced cost criterion links the dual variable π_{s_u} associated to each sensors’ battery constraint (Eq. 6.2) to the cost of using a sensor in a new partition. Hence, a new profitable partition can be computed by maximizing the function:

$$1 - \sum (y_{s_u E_s j} E_s + y_{s_u E_r j} E_r + y_{s_u E_i j} E_i) \pi_{s_u} \quad (6.4)$$

Then, if a column with a positive reduced cost exists, indicating that it may have potential for increasing the lifetime, the column is added to the set Ω' (*i.e.*, an additional column in the model (6.1-6.3)) and the process starts again. Otherwise, the CG process finishes. Furthermore, if an exact approach is used to solve PS, the optimality of the current RMP solution can be proven.

Considering that the model (6.1-6.3) is linear and may be efficiently solved by state-of-the-art solvers, leading to improve the network lifetime those interesting partitions leading to longer network lifetime. With this purpose, the use of a constraint programming approach to solve PS and identify interesting partitions is proposed. Additionally, to accelerate CG and reduce the computational time required to solve the problem through CG, an evolutionary algorithm (EA) is presented. Finally, it is shown that it is worth providing a hybrid exact approach by combining the proposed methods into a single framework that sequentially uses EA and CP, when EA is unable to find a new interesting solution.

6.3.1 A constraint programming model for the pricing subproblem

In this section, some definitions from constraint satisfaction problems (CSP) are briefly introduced. Then, the variables and constraints used to model PS, and the search strategy used to solve it are described. For a more complete review of the theory behind CSP the reader may be referred to Rossi *et al.* [Rossi et al., 2006].

6.3.1.1 Constraint programming definitions

Constraint Satisfaction Problems are widely used to model many Artificial Intelligence issues and combinatorial problems. Given a set of variables, a domain of possible values for each variable, and a conjunction of constraints, where each constraint is a relation defined over a subset of the variables and is used to limit the combination of values that the variables can take, a Constraint Satisfaction Problem (CSP) is to find a consistent assignment of values from the domains to the variables so that all the constraints are satisfied simultaneously [Mackworth, 1977]. Formally defined, a CSP is a triplet $(\mathcal{X}, \mathcal{D}, \mathcal{C})$, where $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ is a set of variables, $\mathcal{D} = \{D(x_1), D(x_2), \dots, D(x_n)\}$ is a set of domains corresponding to the different values that can be assigned to the variables \mathcal{X} , and \mathcal{C} is a set of constraints established between the variables \mathcal{X} .

Definition 1 The **domain** of a variable x , denoted by $D(x)$, is the set of values that can be assigned to x . \underline{x} and \bar{x} are respectively the lower and upper bounds of $D(x)$.

Definition 2 A **constraint** $C_i \in \mathcal{C}$ defined on the subset of m variables $\mathcal{X}(C_i) = \{x_{i_1}, x_{i_2}, \dots, x_{i_m}\}$ is a subset of $D(x_{i_1}) \times D(x_{i_2}) \times \dots \times D(x_{i_m})$. It determines the m -tuples of values which can be assigned to variables $x_{i_1}, x_{i_2}, \dots, x_{i_m}$.

Definition 3 A **solution** of a CSP is a set of assignments of values to variables, $\{(x_1, a_1), (x_2, a_2), \dots, (x_n, a_n)\}$, with $i \in \{1, \dots, n\}, a_i \in D(x_i)$, that satisfies all constraints in \mathcal{C} .

Classically, CSP manipulate integer values in integer domains for integer domain variables; however, set and graph variables have been recently introduced [Dooms et al., 2005, Régim, 2004]. This chapter considers two specific kinds of constraints. The first one is related to the occurrence of values in integer domain variables. The second one concerns the formation of trees, and is used for possible graph values in graph domain variables based on two set variables.

Definition 4 A **Global Cardinality Constraint** is a constraint C on \mathcal{X} , where $\mathcal{X} = \{x_1, \dots, x_n\}$, in which each integer value $a_i \in \mathcal{A}$, where $\mathcal{A} = \{a_1, \dots, a_p\}$, is associated to an integer domain variable $O_i \in \mathcal{O}$, where $\mathcal{O} = \{o_1, \dots, o_p\}$, and $T(C) = \{t \text{ such that } t \text{ is a tuple on } \mathcal{X}(C), \forall a_i \in \mathcal{A} : \underline{o}_i \leq \#(a_i) \leq \bar{o}_i\}$. It is denoted by $globalCardinality(\mathcal{X}, \mathcal{A}, \mathcal{O})$.

It is possible to represent the domain $D(x)$ of a set variable x in a compact fashion by specifying two sets of elements [Gervet, 1997]: the elements that must belong to

the set assigned to x (which we call the lower bound \underline{x}) and the set of elements that may belong to this set (the upper bound \bar{x}). The domain itself has a lattice structure corresponding to the partial order defined by set inclusion.

A graph can be seen as two sets V and E with an inherent constraint specifying that E includes or is equal to $V \times V$. The domain $D(G)$ of a graph variable G is specified by two graphs: a lower bound graph \underline{G} and an upper bound graph \bar{G} , such that the domain is the set of all subgraphs of the upper bound which are supergraphs of the lower bound. For a better understanding of graph variable, the reader may be referred to section 3.10 in [Régim, 2011].

Definition 5 An **anti-tree** $AT = (X, Y)$ is a connected digraph where every node $v \in X$ has exactly one successor $w \in X$, one root $r \in X$ with no successor and for each node $v \in X$, there exists a path from v to r .

Definition 6 An **anti-tree constraint** is a constraint with an input graph variable $G = (V, E)$ that assumes that an anti-tree exists on G .

While solving the problem, filtering rules from the anti-tree constraint will remove arcs from \bar{G} and decisions that add arcs to \underline{G} will be applied until the Graph Variable is instantiated, *i.e.*, when $\underline{G} = \bar{G}$

6.3.1.2 Constraint model

Now reconsider the graph $G(\mathcal{N}, \mathbb{E})$, where $\mathcal{N} = \mathcal{K} \cup \mathcal{S} \cup \{S_0\}$ as defined in Section 6.2. An extra layer $L = \{L_1, \dots, L_q\}$ of nodes is introduced, where $q = \lceil \alpha |\mathcal{K}| \rceil$. To model the PS problem as a CSP to find appropriate P_j , we introduce a directed graph variable $GV(NV, AV)$ that is used to represent the underlying solution. $GV(NV, AV)$ is derived from the graph $G(\mathcal{N}, \mathbb{E})$ in a directed version (by adding one arc from u to v and one from v to u when an edge $e(u, v) \in \mathbb{E}$). NV is a subset of nodes activated in the solution and AV is the subset of arcs used for connectivity. The set of nodes $L \cup \{S_0\}$ is mandatory and always belongs to NV .

Additionally, the following integer vectors variables are used:

- $Succ$ of size $|NV|$ where $D(Succ_u) = \{1, \dots, |NV|\}$ represents the index of the neighbor of node u ($Succ$ is compound of successors from extra nodes in L , successors from node targets \mathcal{K} and successors from sensor nodes \mathcal{S} and $\{S_0\}$, $Succ = SuccL \cup SuccK \cup SuccS \cup SuccS_0$)
- $DegK$ of size m where $D(DegK_u) = \{0, \dots, n\}$ represents the incoming degree on node u in \mathcal{S} from \mathcal{K}
- $DegS$ of size m where $D(DegS_u) = \{0, \dots, m\}$ represent the incoming degree on node u in \mathcal{S} from \mathcal{S}

Additionally a real variable vector $Cost$ of size m is introduced, where $D(Cost_u) = \{0, \dots, \pi_u\}$ is used to represent the dual variable value associated to node u in RMP.

Finally a variable $TotalCost$ sums up all the $Cost$ variables. We fix the upper bound of $TotalCost$ to 1 to force the CSP to produce only profitable columns (see Eq. 6.4). In order to solve the problem we can use the following constraint model:

Model CSP : Pricing Subproblem with Constraints

$$anti-tree(GV) \quad (6.5)$$

$$sumWeight(GV, Cost, TotalCost) \quad (6.6)$$

$$DegK(u) > 0 \iff Cost(u) = E_s \pi_u \quad \forall u \in \mathcal{S} \quad (6.7)$$

$$DegS(u) > 0 \implies Cost(u) \geq E_r \pi_u \quad \forall u \in \mathcal{S} \quad (6.8)$$

$$DegS(u) + DegK(u) = 0 \iff Cost(u) = E_i \pi_u \quad \forall u \in \mathcal{S} \quad (6.9)$$

$$globalCardinality(SuccS, S, DegS) \quad (6.10)$$

$$globalCardinality(SuccK, S, DegK) \quad (6.11)$$

$$channeling(GV, Succ) \quad (6.12)$$

Constraint (6.5) enforces GV to be an anti-tree. In order to consider the filtering rules associated, we use a similar algorithm to the one described by Lorca [Fages and Lorca, 2011] and Unsworth [Prosser and Unsworth, 2006] where the number of trees is fixed to be 1. Complexity of Generalized Arc Consistency (GAC) of such a constraint is known to be enforced in $\mathcal{O}(|NV| + |AV|)$.

The constraint (6.6) computes the overall sum of the costs of all the nodes involved in \mathcal{S} . It is referred as $TotalCost$, and is restricted to be less than or equal to 1 to ensure that only columns that are profitable for RMP are generated. Only Bound Consistency on $TotalCost$ is necessary and the associated scalar complexity is linear in $\mathcal{O}(|\mathcal{S}|)$.

The constraints (6.7-6.9) describe the three different energy consumption conditions of a node, with associated rates E_s , E_r and E_i respectively. The remaining Constraints (6.10-6.12) are associated to the computation of the cost for source and relay sensors. In Constraint (6.12), $Succ$ variables are derived from GV via a standard channeling constraint used to maintain the domain of variable $Succ$ equivalent to the neighbors of nodes in GV (*i.e.*, $v \in D(Succ_u) \iff v \in NV, e(u, v) \in AV$). Then, two Global Cardinality Constraints are used to compute the number of occurrences for node u in \mathcal{S} from respectively the neighbors of \mathcal{S} (6.10) and \mathcal{K} (6.11) (*i.e.*, incoming degree from \mathcal{S} and from \mathcal{K}).

Constraints (6.5) and (6.6) could be aggregated in a single global node weighted spanning tree constraint where GAC propagator can be developed in the spirit of what have been done for the edge weighted spanning tree constraint in [Régim et al., 2010].

6.3.1.3 Search strategy

The search adopted uses a standard *max regret* strategy on *Cost* variables, where the regret is the difference between the smallest value ($E_i\pi_u$) and the next value ($E_r\pi_u$) of the variables. The biggest regret (and first in enumeration) occurs for the most penalized nodes. Value assignments start from minimum value (*i.e.*, trying to deactivate the node). A *reachability propagator* inside the tree constraint ensures that nodes necessary for keeping connectivity will not be removed between target to sink from the final solution. Behavior is quite effective as the median length to find a new column is less than the total number of nodes (*i.e.*, propagation automatically fix last nodes and very few backtracks occurs).

6.3.2 An evolutionary algorithm to boost up CG

The use of heuristic methods is intended to provide near optimal solutions for PS at a lower computational cost than with exact methods. In this way, it is possible to approach near-optimal solutions for RMP efficiently. In the context of WSN optimization, the use of EA has been largely adopted as a strategy to extend the lifetime [Chen et al., 2010, Jia et al., 2008, Lai et al., 2007, Ting and Liao, 2010]. However, most of these approaches do not consider the global objective of maximizing network lifetime but are used to maximize the number of disjoint subsets of sensors. Evolutionary algorithms have been also efficiently combined with CG to solve difficult lifetime optimization problems in WSN. In this chapter, we adopt an EA as a strategy to identify profitable valid partitions based on the reduced cost criterion.

The first step in EAs is the creation of an initial population of solutions for the problem, and then to have this population evolve along multiple generations. The idea is to improve the solutions taking the best characteristics found in selected parents, and keep this process running until a stopping criterion is reached and an appropriate solution is obtained. This chapter considers the inclusion of an additional improvement operator which is used to reduce the total energy consumed for a sensor subset while maintaining the coverage and connectivity constraints. The use of such an operator is intended to improve the quality of the solutions obtained through the proposed EA. The general components of the proposed EA are presented below:

Chromosomes: By following the structure used to represent columns in the model (6.1-6.3) and considering the fact that the cost associated to the network structure is located on the nodes, it is proposed to represent the solutions by using a vector $Z_j = [z_{s_1}, z_{s_2}, z_{s_3}, \dots, z_{s_m}]$ where $z_{s_u} \in \{E_i, E_r, E_s\}$ indicates the energy consumption rate of sensor s_u according to the partition P^j . By using this representation, the structure (and the coefficients) of the column can be easily derived by considering the energy consumption rates associated to a sensor in a given partition P_j .

The example in Figure 6.2 can be used to illustrate the encoding. It represents a valid partition providing full target coverage where $\mathcal{S}_s^j = \{s_5, s_7\}$, $\mathcal{S}_r^j = \{s_1, s_3\}$ and $\mathcal{S}_i^j = \{s_2, s_4, s_6\}$. By considering the energy consumption rate associated to each

operation mode, it is possible to represent this solution by using the chromosome $Z_j = [E_r, E_i, E_r, E_i, E_s, E_i, E_s]$. It can be seen that the structure of the induced tree is not included in the encoding scheme; furthermore, different trees (with the same reduced cost) can be obtained from a single partition.

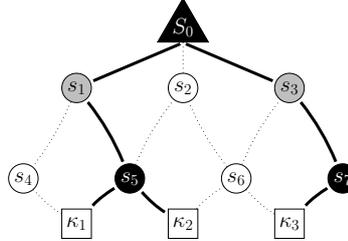


Figure 6.2 – Representation of a valid partition in a simple network.

Initial population: An initial population is generated in a random way by considering only valid partitions inducing a tree rooted at the sink. Previous to generate one solution all sensors nodes are assumed to belong to the subset of idle sensors. Then, in order to generate a random solution, a list of candidate nodes containing the sensors within the communication range of the sink are first considered, and one of them is randomly selected to operate as a source sensor or as a relay. Next, a new list of candidates is created by considering the sensors in idle state that are in the neighborhood of active sensors (or to the sink). Each time that a sensor is added to the group of \mathcal{S}_j^s , the targets that can establish communication with it are added to the set K^j . The process is repeated until $|K^j| \geq \lceil \alpha \mathcal{K} \rceil$.

Fitness: At each iteration of the EA, solutions are evaluated in terms of the objective function that evaluates the reduced cost criterion. The maximization of reduced cost criterion corresponds to the minimization of:

$$\sum_{s_u \in \mathcal{S}} (E_s y_{s_u E_s j} + E_r y_{s_u E_r j} + E_i y_{s_u E_i j}) \pi_{s_u} \quad (6.13)$$

Then, the objective of the EA is to identify the network structure with minimal cost.

Selection: In order to generate a new population, the use of a linear ranking selection procedure is proposed [Baker, 1987]. It consists in allocating a selection probability to the chromosomes based on their rankings within the population. The purpose of this scheme is to increase diversity by permitting some medium-low quality solutions to participate in the creation of a new offspring of child solutions obtained from the parents. This is done to avoid the evolution to be focused in a small portion of the solution space. Moreover, it can be used as an additional improvement to accelerate

the convergence of the CG by returning a diverse set of columns, with active sensors contributing to different constraints, at each iteration of the CG algorithm [Alfandari et al., 2013, Castaño et al., 2014, Mounsla et al., 2010].

Let $F_1 = [f(Z_1), f(Z_2), f(Z_3), \dots, f(Z_{|\mathcal{P}|})]$ be a vector containing the objective function of the solutions in the population and let $|\mathcal{P}|$ denote the size of the population. Let $\rho_i \in \{1, 2, \dots, |\mathcal{P}|\}$ be the ranking of the chromosome i within the population based on the reduced cost criterion, where 0 is assigned to the best ranked solution. Then for each element in the population, the probability of being selected is assigned by using their rank. The probability Θ_i of selecting a parent $i \in \mathcal{P}$ can be computed as:

$$\Theta_i = 2 \frac{|\mathcal{P}| - (\rho_i - 1)}{|\mathcal{P}|(|\mathcal{P}| + 1)} \quad (6.14)$$

And,

$$\sum_{i \in \mathcal{P}} \Theta_i = 1 \quad (6.15)$$

Then the selection is obtained by performing a classical roulette wheel selection within the elements of the population.

Crossover: The evolutionary algorithm generates iteratively new solutions to be inserted in the population. In order to generate a new population that will replace the worst elements of the previous generation, the proposed genetic algorithm applies a Bernoulli crossover strategy to each of the solutions that were selected through the selection operator [Jong and Spears, 1992, Spears, 1991]. In this strategy a new pair of solutions is generated by swapping the elements between the two parents a and b with a probability given by $P_a = \frac{Fit_a}{Fit_a + Fit_b}$. As expected, the use of such an approach can lead to infeasible solutions. As a consequence, the use of repairing procedures is intended to allow new solutions to be included in the new generation provided they are not already part of it.

Feasibility repair: After the crossover procedure between parent solutions is performed, unfeasible child solutions could be obtained in the new offspring. Hence, new solutions must be repaired to become valid partitions, columns, for model (6.1-6.3). In order to do this, two repairing strategies are proposed to be used depending on the kind of unfeasibility observed in the solution. First, a *coverage* repair is used when the evaluated solution does not reach the required level of target coverage and additionally source sensors are required. Second, a *connectivity* repairing procedure is proposed to add some extra sensors to restore network connectivity.

In order to keep a diverse population, both procedures are designed in such a way that the decisions taken to repair the solution are non-deterministic. Consequently, similar (or even duplicate) solutions are not necessarily repaired in the same way and the feasibility repair procedure can be seen as a sort of mutation procedure.

Algorithm 6.1 presents the repairing procedure. It takes the current level of coverage as input (line 2) and iteratively activates as source some of the the sensors that are

inactive or operating as relays ($S_{av} = S_{inactive} \cup S_{relay}$). This process is repeated until the required level of coverage is reached (line 5). The contribution of inactive and relay sensors is evaluated by the ratio between activation cost and the additional coverage they provide (lines 6-16). Then, this piece of information is used to calculate an activation probability P_{sel} (lines 18-20) that is used to randomly select the new sensor s_{sel} that will be used as a source (lines 21-28).

Algorithm 6.1: Coverage repairing procedure

```

1  $S_s \leftarrow \{s_u \in Z : z_{s_u} = E_s\}$ 
2  $T_{unc} \leftarrow \{\kappa_i \in \mathcal{K} : \nexists e(\kappa_i, s_u) \forall s_u \in S_s\}$ 
3  $K^j \leftarrow \mathcal{K} \setminus T_{unc}$ 
4  $S_{av} \leftarrow S \setminus S_s$ 
5 while  $|K^j| < \lceil \alpha \mathcal{K} \rceil$  do
6   for  $s_u \in S_{av}$  do
7      $Cov_{s_u} \leftarrow \{\kappa_i \in T_{unc} : \exists e(\kappa_i, s_u)\}$ 
8     if  $|Cov_{s_u}| > 0$  &  $s_u \in S_r$  then
9        $Ind_{s_u} \leftarrow \frac{\pi_{s_u}(E_s - E_r)}{|Cov_{s_u}|}$ 
10    else
11      if  $|Cov_{s_u}| > 0$  then
12         $Ind_{s_u} \leftarrow \frac{\pi_{s_u} E_s}{|Cov_{s_u}|}$ 
13      else
14         $S_{av} \leftarrow S_{av} \setminus s_u$ 
15      end
16    end
17  end
18  for  $s_u \in S_{av}$  do
19     $P_{sel_{s_u}} \leftarrow Assign\_selection\_probability(Ind_{s_u})$ 
20  end
21   $x \leftarrow U[0, 1]$ 
22   $P_{acc} \leftarrow 0$ 
23  while  $P_{acc} \leq x$  do
24     $P_{acc} = P_{acc} + P_{sel_{s_u}}$ 
25    if  $P_{acc} > x$  then
26       $s_{sel} \leftarrow s_u$ 
27    end
28  end
29   $S_{av} \leftarrow S_{av} \setminus s_{sel}$ 
30  for  $\kappa_i \in \mathcal{K} \mid \exists e(\kappa_i, s_{sel})$  do
31     $T_{unc} \leftarrow T_{unc} \setminus \kappa_i$ 
32     $K^j \leftarrow K^j \cup \kappa_i$ 
33  end
34   $S_s \leftarrow S_s \cup s_{sel}$ 
35 end

```

In a similar way, disconnected solutions are repaired by using a parameter-free procedure (Algorithm 6.2). As it never changes, the shortest path between two sensor

nodes (measured as the number of hops required to send information from one to the other) is calculated at the beginning of the CG process by using the Floyd-Warshall algorithm [Floyd, 1962, Warshall, 1962] and it is never calculated again; however, it is used each time that connectivity needs to be repaired. In the EA, when a solution that does not meet the connectivity requirements is found, the different connected components CC are computed (line 2). For each pair of elements being in separated components the distance is retrieved and used to allocate the probability of repairing through the shortest path between the two nodes (lines 9-15). Then, a pair of sensors s_u, s_v is randomly selected, based on the probabilities computed (lines 16-23), and the sensors in the path $Path_{s_u, s_v}$ connecting them are activated as relay nodes (lines 24-27). In order to evaluate if the current solution is already feasible, connected components are calculated again (line 28). If more than one connected component is detected in the current solution, the whole process is repeated until full connectivity is reached. The process might be stopped as well if the required level of coverage is achieved by the connected component containing the base station. In the latter case, sensors not in this component must be removed.

Population improvement: Once two feasible solutions are obtained, a post-processing is performed to improve them. As it could be expected, the application of crossover and repairing procedures can lead to solutions consuming more energy than necessary. Then, the proposed improvement procedure is intended to reduce the energy consumption by decreasing the consumption rate associated to the source or relay sensors. The proposed procedure applies a best-improvement strategy, and checks if each active sensor can be moved to the set of idle sensors whilst keeping the solution feasible. If the partition remains feasible, the procedure accepts this movement and continues the process evaluating other sensors. Otherwise, it attempts to turn each source sensor into a relay. The improvement process finishes when all the active sensors have undergone an attempt for a lower power consumption role. In order to avoid premature convergence, the improved solutions are added to the new offspring to replace the weakest members only if they are not already part of it, *i.e.*, no duplicated elements are allowed.

6.3.3 Hybrid CG+EA+CP approach to maximize network lifetime

Both the proposed CP and EA approaches used to solve PS have their pros and cons. While the use of exact approaches to solve PS allows to guarantee that the solution obtained through the use of CG is optimal, it might lack of efficiency in computing interesting columns for PS and can be impractical in large scale applications. By contrast, although EA is expected to be more efficient in computing interesting columns, it neither guarantees that these are optimal for PS nor allows to guarantee that the CG process stops when it reaches the optimal solution.

In order to take advantage of the best features of both approaches, they are combined into the CG framework. Our proposal is to apply both methods sequentially to solve PS. First, EA is used to obtain new interesting columns in low computational time and to return them to model (6.1-6.3) when they have an interesting reduced cost. If no

Algorithm 6.2: Connectivity repairing procedure

```

1  $S_{av} \leftarrow \mathcal{S} \setminus \mathcal{S}_s \cup \mathcal{S}_r$ 
2  $CC = \{cc_1, cc_2 \dots cc_n\} \leftarrow Identify\_Connected\_Components(\mathcal{S}_s, \mathcal{S}_r)$ 
3 for  $(s_u, s_v | s_u \in cc_i, s_v \in cc_j, i \neq j)$  do
4    $Path_{s_u s_v} \leftarrow Shortest\_Path(s_u, s_v)$ 
5 end
6  $d_{max} \leftarrow \max\{|Path_{s_u s_v}| : s_u \in cc_i, s_v \in cc_j, i \neq j\}$ 
7  $d_{min} \leftarrow \min\{|Path_{s_u s_v}| : s_u \in cc_i, s_v \in cc_j, i \neq j\}$ 
8 while  $|CC| > 1$  do
9   for  $(s_u, s_v | s_u \in cc_i, s_v \in cc_j, i \neq j)$  do
10     $Ind_{s_u s_v} \leftarrow \frac{d_{max} - |Path_{s_u s_v}|}{d_{max} - d_{min}}$ 
11   end
12
13   for  $(s_u, s_v | s_u \in cc_i, s_v \in cc_j, i \neq j)$  do
14     $P\_sel_{s_u s_v} = \frac{Ind_{s_u s_v}}{\sum_{s_l \in cc_i, s_m \in cc_j, i \neq j} Ind_{s_l s_m}}$ 
15   end
16    $x \leftarrow U[0, 1]$ 
17    $P_{acc} \leftarrow 0$ 
18   while  $P_{acc} \leq x$  do
19      $P_{acc} = P_{acc} + P\_sel_{s_u s_v}$ 
20     if  $P_{acc} > x$  then
21        $Path_{sel} \leftarrow Path_{s_u s_v}$ 
22     end
23   end
24   for  $s_{sel} \in S_{av} | s_{sel} \in Path_{sel}$  do
25      $S_{av} \leftarrow S_{av} \setminus s_{sel}$ 
26      $\mathcal{S}_r \leftarrow \mathcal{S}_r \cup s_{sel}$ 
27   end
28    $CC \leftarrow Identify\_Connected\_Components(\mathcal{S}_s, \mathcal{S}_r)$ 
29 end

```

interesting solution is found with this heuristic approach, EA is replaced as column generator by an exact approach based on CP used to check if either the current solution for model (6.1-6.3) is optimal or a new solution can be obtained. In the former case the CG process stops and the solution is optimal. Otherwise, a new iteration is performed and EA is used again to try to identify interesting columns for model (6.1-6.3).

6.4 Computational experiments

In order to evaluate the performance of the proposed approaches, experiments are carried out on a set of 240 instances. The instances are characterized by a number of sensors $|\mathcal{S}|$ in $\{100, 200, 300, 400, 500\}$ and each instance is evaluated with a number of targets $|\mathcal{T}|$ in $\{15, 30\}$. Both, sensors and targets are deployed in a 500×500 area, sensor have a communication range R_c of 125 and a sensing range R_s in $\{100, 125\}$.

The energy consumption rate associated to a sensor acting as a source is $E_s = 1$, as a relay is $E_r = 0.8$ and idle is $E_i = 0$. Finally, three values for $\alpha \in \{0.7, 0.85, 1.0\}$ are considered to denote the fraction of the set of targets that must be covered for a valid partition. The experiments are performed on a Intel Xeon 2.8Ghz workstation with 8GB of RAM. We use Gurobi 5.6 to solve RMP and Choco 3.2 to solve the PS via constraint programming. If the proposed methods have not been able to find the optimal solution of the problem within a time limit of 3600 second, the procedure stops and the best solution is reported. At the first iteration, RMP is initialized with a single column in which all sensors are operating at a maximum energy consumption rate (*i.e.*, $S_s^0 \leftarrow \mathcal{S}$).

Computational experiments indicate that the convergence of CG+EA might be affected by the number of columns returned at each iteration of CG. In order to accelerate the convergence of CG we return up to n_{max} columns at each iteration selected from the population. In this way, it is possible to take advantage of some characteristics of EA, as diversity, to improve convergence to a (near-)optimal solution [Moungla et al., 2010]. Figure 6.3 displays a boxplot indicating the reduction (%) on the number of iterations obtained when different values of n_{max} are used compared to the case when a single column is returned. As observed, the convergence is gradually improved when more than a single column is returned to RMP; nonetheless, according to our experiments, in general the difference in computational time is less noticeable when more than 20 columns are returned (see Figure 6.4). In this chapter we adopt this intensification strategy by returning up to $n_{max} = 20$ at each iteration of CG.

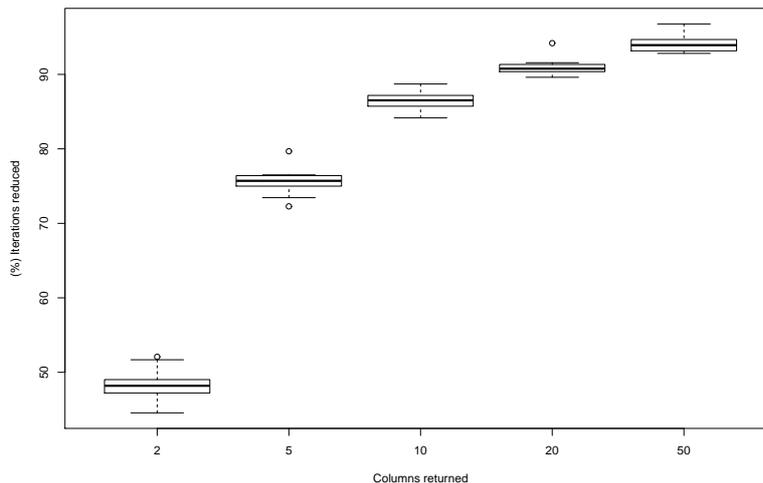


Figure 6.3 – Reduction of the number iterations (%) by returning several columns throughout CG+EA+CP.

Figure 6.5 depicts the typical behavior observed for the evolution of the objective function throughout CG+EA+CP iterations. The results seems to indicate that by

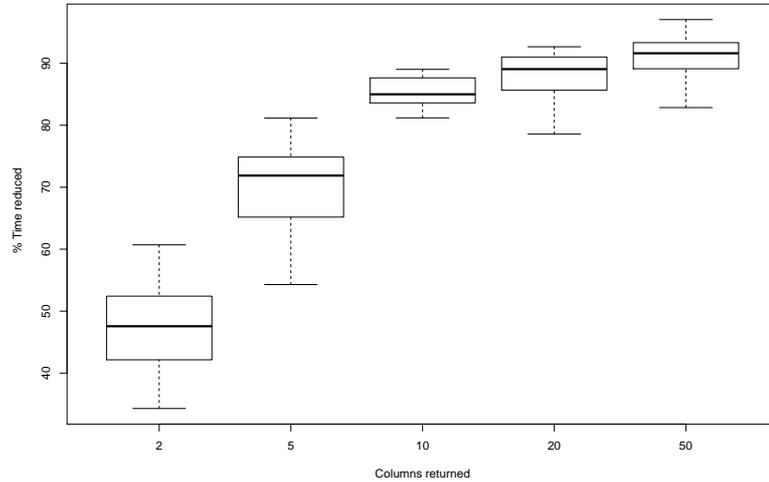


Figure 6.4 – Reduction on the CPU time (%) by returning several columns throughout CG+EA+CP.

returning several columns it is possible not only to accelerate the convergence towards the optimal solutions but also to provide interesting suboptimal solutions in a low number of iterations. In general, this reduction is also appreciated in the CPU time required to compute such solutions. Hence, it might indicate that even if the proposed approach is unable to find the optimal solution, it is still able to compute interesting solutions.

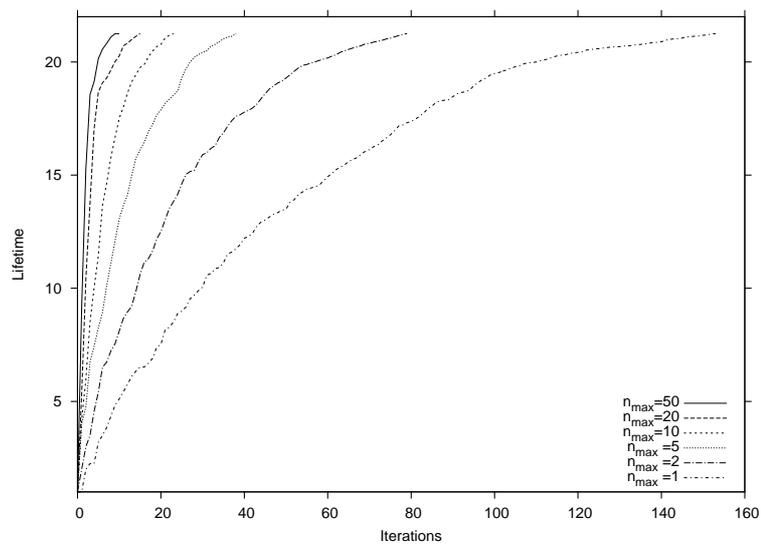


Figure 6.5 – Objective function evolution throughout CG iterations.

We evaluate the proposed approaches in terms of their ability to compute an optimal solution for α -CMLP-MR. Table 6.1 divides the analysis of the instances into groups with similar characteristics to evaluate the performance of the proposed approaches according to them. The columns with the label CG+CP indicate the results obtained when only the CP is used to compute valid partitions to be included in RMP. In the same way, the columns labeled CG+EA and CG+EA+CP display the results obtained when only EA or EA and sequentially CP are used respectively. In the latter case the approach indicates whether there exists a new interesting partition or the current solution is optimal. As it could be observed, CG+CP is an efficient method useful to compute a schedule that maximizes network lifetime. It is able to find the optimal solution for 76% of the total set of instances. Indeed, this result is largely improved when the analysis are limited to the instances that require 100% of coverage ($\alpha = 1$), *i.e.*, 95% of the instances are solved to optimality within the time limit. The performance is clearly reduced when partial coverage is allowed where the method achieves 65% and 40% of the optimal solutions for $\alpha = 0.7$ and $\alpha = 0.85$ respectively. This might be partially explained by the increase in the size of the space search as a consequence of the relaxation of the coverage requirements which leads to a bigger computational effort at each CG iteration.

Computational experiments show that CG+CP approach is able to find the optimal solution for about nearly 67 % of the whole set of instances. This result is improved through CG+EA that is able to find better solutions than CG+CP and obtains the optimal solutions for 80% of the instances in a lower computational time; however, it is not able to prove that these solutions are optimal. Hence, CG+EA+CP might be conceived as the best strategy to address α -CMLP-MR. By using this approach, the results obtained by CG+CP and CG+EA are improved in terms of solution quality but most importantly in terms of computational time. The combined effect of both approaches allows to produce high quality solutions in a low time and allows to demonstrate that solutions are optimal such that it is an exact solution strategy. As EA is a heuristic approach, a standalone method based on it will probably get stuck in suboptimal solutions. Consequently, the collaboration within the CG framework of EA and CP allows to exchange easily between the two methods.

We evaluate the performance of CG+EA and CG+EA+CP against the standalone application of the CG+CP approach to address pricing subproblem for the set of instances demanding a 100% level of coverage ($\alpha = 1$). Results indicate that both CG+EA and CG+EA+CP are able to find all the optimal solutions for this set; similarly, CG+CP is able to compute up to 97% of these solutions and the remaining 3% of instances correspond to the instances with the largest number of sensor nodes.

Table 6.2 presents the average CPU time obtained with the proposed approaches. The table discriminates between those instances for which the method was able to compute the optimal solution within the time limit. According to the experiments, CG+EA can be used to reduce the computational time required to approach optimal solutions faster than CG+CP. Then, the reinforcement with the use of CP allows to guarantee that the solution obtained for α -CMLP-MR is optimal. The computational experiments seem to indicate that CG+EA and CG+EA+CP might be used to compute ef-

Instances		$\alpha = 0.7$			$\alpha = 0.85$			$\alpha = 1$		
$ S $	$ K $	CG+CP	CG+EA	CG+EA+CP	CG+CP	CG+EA	CG+EA+CP	CG+CP	CG+EA	CG+EA+CP
100	15	6	7	7	2	6	6	8	8	8
	30	5	6	6	3	4	4	8	8	8
200	15	6	7	7	4	5	5	8	8	8
	30	5	7	7	3	5	5	8	8	8
300	15	8	8	8	6	8	8	8	8	8
	30	8	8	8	5	8	8	8	8	8
400	15	6	6	6	4	6	6	8	8	8
	30	5	6	6	3	5	6	8	8	8
500	15	2	3	3	2	2	2	6	8	8
	30	2	3	3	0	2	2	6	8	8
Total		53	61	61	32	51	52	76	80	80

Table 6.1 – Number of optimal solutions computed by each method in the different groups of instances

ficiently interesting solutions for the evaluated problem. It is observed that it computes a higher number of optimal solutions in a lower computational time. Experimental analyses indicate that when CG+EA cannot compute the optimal solution it may be because of the two following reasons: (i) the EA is not able to compute an interesting new column or (ii) the lifetime increases only marginally at each iteration and convergence to the optimal solution is slow. In addition, the computational experiments using the CP approach suggest that as CG approaches the optimal solution, the computational effort required to solve PS might become too big and, sometimes, the CG process get stuck at a single iteration. As a consequence, even the single iteration required to prove optimality may consume an important amount of time.

Finally, Table 6.3 compares the results in term of lifetime for the different groups of instances. The results indicate that, adding more targets to a particular network while maintaining reduced levels of coverage (*e.g.*, $\alpha \in \{0.7, 0.85\}$) do not necessarily decreases the lifetime. It appears to be related with the fact that, having more targets, it is possible to provide alternative network configurations that avoid the coverage of the set of weakly covered targets that lead to lifetime shortages. As expected, the computational results indicate that, by allowing lower levels of coverage, lifetime can be largely extended. This phenomenon is appreciably notorious when network lifetime is bounded by the less covered targets.

6.5 Conclusions and future work

In this chapter the maximum network lifetime problem in wireless sensor networks with coverage and connectivity constraints is discussed. Sensors having up to three operation modes adopting different energy consumption profiles depending on the tasks they perform in the network are considered. In order to solve the problem, an efficient exact approach based on column generation is proposed. The method divides the problem into two decisions. First, decisions related to the allocation of schedules for the operation of each sensor node. Second decisions that guarantee that during lifetime the schedule respects the coverage and connectivity constraints. Scheduling decisions are taken by solving a linear program defined over an exponential number of variables that can be efficiently solved by using column generation. Nonetheless, to efficiently identify

α	$ S $	$ K $	Non-optimal			Optimal			
			CG+CP	CG+EA	CG+EA+CP	CG+CP	CG+EA	CG+EA+CP	
0.7	100	15	3600.0	82.2	3600.0	35.4	15.1	15.9	
		30	3600.0	213.0	3600.0	21.1	9.8	10.5	
	200	15	3600.0	441.4	3600.0	70.1	51.0	70.3	
		30	3600.0	576.3	3600.0	71.3	29.7	30.7	
	300	15	-	-	-	457.7	29.9	31.0	
		30	-	-	-	343.6	31.3	32.3	
	400	15	3601.6	1477.5	3600.0	768.0	52.7	54.0	
		30	3566.7	2687.5	3600.0	1003.3	149.4	151.0	
	500	15	3600.0	3600.0	3600.0	2600.5	243.0	244.7	
		30	3600.0	3600.0	3600.0	2857.4	346.3	348.3	
	0.85	100	15	3600.0	186.3	3600.0	7.1	29.6	30.2
			30	3600.0	250	3600.0	10.1	69.1	69.8
200		15	3600.0	584.3	3600.0	44.3	63.9	65.2	
		30	3600.0	873.1	3600.0	298.1	59.8	60.6	
300		15	3600.0	-	-	296.8	67.8	68.9	
		30	3600.0	-	-	446.5	174.3	175.6	
400		15	3600.0	2204.5	3600.0	1302.0	471.0	472.3	
		30	3600.0	3600.0	3600.0	1027.1	533.7	534.8	
500		15	3600.0	3600.0	3600.0	1714.2	163.3	165.0	
		30	3600.0	3600.0	3600.0	-	589.5	591.5	
1.0		100	15	-	-	-	3.4	10.9	17.6
			30	-	-	-	4.1	22.5	31.6
	200	15	-	-	-	49.2	18.8	26.6	
		30	-	-	-	39.7	24.7	32.5	
	300	15	-	-	-	290.0	34.3	40.9	
		30	-	-	-	134.0	38.2	49.3	
	400	15	-	-	-	437.8	45.3	52.8	
		30	-	-	-	490.4	81.6	85.5	
	500	15	3600.0	-	-	1279.8	186.9	197.3	
		30	3600.0	-	-	1519.8	275.8	298.1	

Table 6.2 – Comparison of computational time (s) required for the proposed approaches

α	$ S $	$ \mathcal{K} $	CG+CP	CG+EA	CG+EA+CP	
0.7	100	15	8.4	8.5	8.5	
		30	8.4	8.6	8.6	
	200	15	19.2	19.8	19.9	
		30	19.2	20.0	20.0	
	300	15	22.8	22.8	22.8	
		30	22.8	22.8	22.8	
	400	15	38.4	38.4	38.4	
		30	34.5	37.3	37.3	
	500	15	49.1	57.6	58.0	
		30	49.0	57.3	57.5	
	0.85	100	15	7.9	8.0	8.1
			30	7.4	7.9	8.0
200		15	16.4	18.9	18.9	
		30	17.1	18.5	18.6	
300		15	22.4	22.6	22.6	
		30	22.4	22.7	22.7	
400		15	33.5	36.1	36.1	
		30	29.7	33.5	34.6	
500		15	38.5	51.2	51.3	
		30	40.4	49.0	43.8	
1.0		100	15	4.8	4.8	4.9
			30	.6	4.6	4.6
	200	15	12.6	12.6	12.6	
		30	10.7	10.7	10.7	
	300	15	17.3	17.3	17.5	
		30	15.0	15.2	15.0	
	400	15	22.6	22.6	22.6	
		30	21.0	21.0	21.0	
	500	15	33.2	33.7	33.7	
		30	30.7	31.2	31.2	

Table 6.3 – Average network lifetime computed through the proposed approaches

network partitions leading to extend the lifetime the use of specialized approaches are devised. In this chapter the use of an Evolutionary Algorithm is proposed to approach (near-) optimal solutions. Then, when it fails, a constraint programming approach is used to confirm that solution is optimal or identify an additional interesting column.

The computational results suggest that the proposed approach is an efficient strategy to compute (near-)optimal solutions for the problem. It is demonstrated that when full coverage of the targets is necessary, the method presents an interesting performance that makes it worth to solve the problem and outperforms other efficient approaches. However, as expected, the performance decreases when partial coverage is allowed, yet it is still able to compute competitive solutions for the problem.

Regarding future lines of research, we consider that the proposed method can be further improved by using sophisticated integer programming and constraint programming approaches taking advantage of the characteristics of the model. Additional studies leading to overcome the convergence of the CG approach applied to WSN optimization can be devised. In future research, a comprehensive model and solution approach able to consider, in a single framework, different models concerning the energy consumption and operational constraints of WSN will be studied.

7

General conclusions and future works

7.1 General remarks

This research is motivated by the energy efficient design of the operation in randomly deployed Wireless Sensor Networks with constrained resources. Different characteristics of the network operation are considered including several models for energy consumption and sensors able to perform different tasks within the network. Special emphasis is put on the connectivity issue which largely increases the complexity of the design problems involved in the energy efficient design of WSN. Additionally, our research has considered several problems in which partial coverage of discrete targets is allowed, so it is possible to take advantage of this characteristic to improve network lifetime. Finally, the coverage and connectivity constraints are considered in WSN with optional capabilities, sensors using energy at different energy consumption rates, that can be used to extend network lifetime.

7.1.1 Thesis summary

In this research it is shown that a general modeling strategy can be used to tackle several problems related with the energy efficient design of WSN. The model, however, is typically defined over an exponential set of variables that might prevent handling even small size networks with a low number of sensors. Consequently, to address the problem this work is mainly focused on the use of column generation (CG) as the strategy adopted to address different versions of the problem. In this way, it is possible to avoid the complete enumeration of all the variables but somehow enumerate only the promising ones to improve the efficiency on the usage of the resources provided by the sensors, specially their limited energy budget.

CG is seen as a general framework to optimize the use of energy in WSN that can be easily adapted to include the different constraints imposed for the network operation and sensors resources. Thus, the CG adopted easily represents the energy consumption scheme for wireless sensor nodes, yet the pricing subproblem has to be modified to consider any characteristic required and to provide columns representing the energy consumption for each node at each specific configuration.

Two decisions are considered when using CG: timing decisions, and topology decisions. The topology decisions involve identifying energy-efficient configurations of the

network (including decisions of role within the network and routing) that can be used to extend network lifetime, and the timing decisions which assign the operational time for those interesting configurations. In this way, the use of CG implies that network design problems are required to be solved iteratively to optimize the use of the energy.

To generate new columns, the CG approach adopted relies exclusively on the energy consumed by the sensors and does not make any assumption on the structure of the area or targets to be covered. This is an obvious consequence of the fact that communications and coverage links are only virtually established through wireless signals and the energy is consumed only in sensor nodes. Hence, the optimal timing must respect the battery constraints of each individual node. Moreover, it can be extended to consider additional sources of energy consumption by including additional characteristics in the pricing subproblem used to define the structure of the network and the operation of the network at any moment

7.1.2 Methodology and findings summary

The results obtained through classical implementations of the CG method, based entirely on the use of MIP models solved through state-of-the-art solvers, show that it is highly limited in scope. The experiments seem to indicate that CG cannot be easily applied to solve problems of higher complexities including those in which connectivity constraints must be enforced. Two main reasons were identified as the causes of this poor performance. On the first hand, the pricing subproblem might be too hard to solve and a big computational effort might be required to generate an interesting column to be included in the restricted master problem. Consequently, the time required to solve a problem up to proven optimality can be huge and CG might be unpractical. On the second hand, the convergence of the method itself can be a problem. It was observed that CG performance is particularly bad when it has to be used to consider problems with partial coverage in which the solution space largely increases in size, and so the number of feasible columns. Hence, the convergence of CG is affected.

Depending on the problem addressed, CG can present slow convergence arising from different sources. In some cases CG presents the typical tail-off effect that imply that it has problems to find the optimal solution of the problem when it is close to it. Thus, the method keeps generating columns in the last iterations that improve the objective function only marginally. In other cases, CG might present a heading-in effect and it produces, for a number of iterations, irrelevant (or bad quality) columns that have a limited effect on the evolution of the objective function and are highly unlikely to be part of the optimal solution, *e.g.* columns having a lot of active sensors. In the problems addressed in this work, this problem can be partially explained by the fact that those columns are densely populated contributing to a lot of constraints, a characteristic that is as undesirable as inefficient in terms of energy consumption.

This research explored different strategies to cope with the problems affecting the performance of the CG used to optimize the energy usage in WSN. The effect of dual variable stabilization and intensification strategies is evaluated to avoid the tail-off effect as well as simple strategies that can be used to address the heading-in effect by avoiding

the construction of highly populated columns. The numerical experiments indicate that better performance of CG can be achieved by using each of these methods at different levels. We observed that dual variable stabilization can be used to reduce the number of iterations required to compute an optimal solution through CG. Moreover, the results might signal that by providing several and diverse columns, stabilization is indirectly obtained, and consequently, an improved CG convergence. In contrast, no evidence indicates that providing initial solutions to initialize the CG brings significant improvements.

Additionally, the performance of the CG can be improved by using efficient methods to address the pricing subproblem. In this case we considered two approaches: (i) the use of specialized exact approaches able to tackle efficiently the pricing subproblem, and (ii) the use of heuristic approaches to approximate quickly solutions for the pricing subproblem and, consequently, interesting (near) optimal solution for the whole problem. In the first case, the use of Constraint Programming and an efficient Benders Decomposition approach were considered. In the latter, the use of metaheuristics is explored. It is used to solve the pricing subproblem and, additionally, to take advantage and produce diverse columns to be returned to the master problem and contribute to improve the performance of CG.

Computational results seems to indicate that CG can be efficiently combined with exact and heuristic approaches; nevertheless, the method requires the use of advanced strategies to deliver its full potential. The energy efficient design of wireless sensor networks implicitly requires being able to address effectively a large variety of network design problems. Thus, it is possible to take advantage of some advances developed in this field to create interesting solution approaches. Nonetheless, some of the derived pricing subproblems might not be neither too common nor deeply explored. Hence, the development of new strategies to face them efficiently is required as they can be computationally demanding as well. It was learned that the inner characteristics of the derived network design problems can be exploited to solve the pricing subproblem and to boost up the CG. Finally, it was observed that it is worth combining some of the methods into a single approach that takes advantage of the diverse enhancements they provide to CG.

Next the limitations of the current study are presented, such that they can be seen as the starting point for future research. Moreover, some potential future work closely related with the research presented throughout the manuscript is introduced and some extensions to different domains as communication networks design.

7.2 Limitations of this study

Some strengths have been presented that indicate why the use of CG is an interesting option should be considered in the optimization of the energy usage in WSN. However, the present study and the technique itself present several limitations that might require to be dealt with to improve the quality of the results found. In the following we highlight some interesting problems that are still open issues in WSN energy-efficient design.

Area coverage vs. Discrete coverage The proposed study does not consider directly area coverage. Some of the solution approaches proposed in this work might be used in several problems in which it is required to satisfy area coverage either by introducing some techniques to transform area coverage into target coverage or by approximating the area coverage using discrete points [Deng et al., 2012, Rebai et al., 2014, Tan et al., 2013]. However, depending on the conditions on the coverage required, it might be necessary either to do a big modification of the method to be adapted or to develop another kind of solution approach for the pricing subproblem that in some cases might imply to solve non-linear problems.

The effect of randomness In this research it is considered that wireless sensors do not fail. However, although the failure rate might be low, it can be necessary to deal with this characteristic of the sensor nodes in order to produce robust and reliable WSN operations schedules. Moreover, it can be necessary to contemplate other sources of randomness at the moment of design the WSN operations. If these characteristics are not considered, the quality of the coverage provided can be affected and so does the reliability of the information collected by the sensor nodes. This problem can be partially managed by guaranteeing that survivable network structures are generated when looking for an interesting network configuration through the pricing subproblem. In this way, it is possible to prevent that a certain level of sensor failures affect largely the coverage objective for which sensors are deployed. As a matter of example, this problem can be partially tackled by considering networks structure that provide either multiple coverage of the targets or guarantee that critical regions are surveyed by several sensors at the same time [Singh et al., 2013].

Multi-purpose wireless sensor networks Throughout the manuscript our analysis was completely focused on wireless sensor networks used to monitor a single phenomenon or variable. Nonetheless, this is not always the case. Sensors might be provided with multiple sensing units that can be used to monitor different variables either at the same time or at different moments depending on the requirements. This characteristic was not considered in our models; however, in certain cases it can be directly included in our solution approaches by considering these variables as different targets to be monitored and including the characteristics associated to the consumption in the models used to solve the problems. Nonetheless, in order to guarantee that the approaches proposed here can be extended to that case, further experimental analysis are required.

Non-homogeneous wireless sensor networks Our experimental analysis were entirely based on the use of homogeneous sensor nodes. Although the general framework used to address the problem, and based on CG, can be extended for those cases, they have not been directly considered in this work. It can be necessary to perform further analysis to include additional characteristics as different initial battery charges or sensors with different sensing units. In some cases, large modifications are not needed and it is enough by indicating which sensor is able to cover which target. Other cases may

require slightly bigger modifications depending of the objective and the capabilities of the WSN; however, an initial approach to solve a wide class of problems is provided in this work.

Centralized vs. Decentralized approaches The CG method applied to solve the problems addressed in this thesis correspond to a centralized solution approach that is intended to be indicated to the sensors after knowing its location and the optimal solutions. Nonetheless, in some cases, practitioners might prefer the use of distributed approaches in which a global decision is not necessary but the operation of the network is defined over decisions taken by the individual nodes that are part of the network. Our approach cannot be easily extended to consider this case, still it can be used to evaluate the performance of distributed approaches in metrics as the lifetime or the quality of coverage provided during the network lifetime. In some cases, it can be possible to establish communication with a distant PC; then, the problem can be solved far away from the network, and the solution can be uploaded to the sensors.

7.3 Perspectives of research

As this study is limited, there exists yet a lot of work that can be done to improve the quality of the analysis proposed and to provide a general method able to deal with generalizations of the proposed methods. Some extensions can be derived almost directly from the basic ideas involved in the proposed approaches. Nonetheless, other cases might require additional considerations to be included. This is especially the case when the use of heuristics is required, as a consequence of the fact that some of the methods proposed are specifically designed to tackle the problems they were intended to solve.

Moreover, it is possible to devise several related problems in different fields that may deserve some attention over the basis of the approaches proposed in this work. Throughout the manuscript we explored slightly interesting network design problems that can be extrapolated to fields such as communication or energy transmission networks, pipeline distribution systems, supply networks design, among others. We propose to evaluate the performance of the techniques presented here in some of those problems. In this way, it might be possible to produce solution approaches not only for the proposed problems (derived as pricing subproblems) but also methods that can be exploited by the CG to solve general versions of the problems addressed in this research.

It was already mentioned that to include all of the characteristics of the networks and to solve the problems through CG it is required to develop efficient strategies to solve node weighted and generalized versions of several network design problems. In some cases the task might not be too complex; however, in most cases we face highly combinatorial problems that may need a big computational effort to be solved or that cannot be solved to proven optimality with current solution approaches. As a consequence, the problems derived from the computational approach adopted have opened the door to study several problems that in some cases have not received too much

attention in previous research. In this section, we present several directions for future research that either were indirectly considered throughout the research or will require to be considered in order to extend the CG solution approach to a more general class of WSN. Initial steps will be addressed towards extensions of the proposed approach such that, gradually, it will be possible to provide a general approach. Some of the related problems that will be addressed in our future research are described below.

Maximum lifetime and general role allocation in WSN The concept of roles introduced in Chapters 5 and 6 can be further extended and generalized to WSN in which sensors play different functions in the network. In this way, it can be possible to consider the direction of a sensor in a wireless directional sensor network as a different role to play that may or may not consume energy at a different rate. If sensors can modify the size of the surveyed area, their communication or sensing ranges, monitor different variables or combine several of these characteristics in a single device it can be assumed that each function represents a different role. Whenever the connectivity issue is also considered, it can be possible to extend the approaches previously described. Thus, we can adopt and extend the previously proposed cutting-planes algorithm to face the pricing subproblem. In this way, the CG algorithm previously described can be directly applied for some of the problems. Moreover, solution approaches based on greedy heuristics for generalizations of the Steiner tree problems in graphs have been already developed. These approaches can be used as the basis to develop efficient heuristics to tackle the pricing subproblems. As optimality is desired, it can be possible to develop versions of the Benders Decomposition and cutting-planes algorithms explored in this work and combine them with heuristic approaches to produce efficient exact approaches.

Scarcely populated wireless sensor networks In certain applications the number of sensors available to monitor a certain phenomenon is not too large and/or all the sensors are required to be active all the time until network dies. If sensors are able to modify the communication range, a typical approach to save energy in this case is to reduce the maximum energy consumption rate for each sensor in such a way that connectivity is still present. Nonetheless, it is still possible to use different network configurations during the network lifetime with the purpose of extending network lifetime. In the former case, previous researches have led to interesting approximation algorithms to approach solutions minimizing the total energy consumed and indirectly the individual energy consumed by sensor nodes. Similar strategies can be adapted to solve the weighted version of these problems that is derived from the adoption of the CG approach. In this way, it can be possible to improve the efficiency in the usage of energy in WSN and further extending the lifetime of the network.

Maximum number of disjoint set k -covers problem In order to extend lifetime, some authors have proposed the use of disjoint subsets that provided a required level of coverage (for example, covering more than k times each target). To solve this problem,

it is necessary to use a different strategy; nonetheless, it can be based on branch and price and cut which means that previously proposed approaches may still be useful at every node of the branch and bound tree. In this way, branching strategies can be combined with any solution method used to solve PS and to provide both new columns and new bounds required for the adoption of branch and bound strategies.

Sink mobility and super capabilities The use of mobile sinks has been demonstrated to be an interesting strategy to improve the lifetime in wireless sensor networks and the efficiency in the use of sensors energy. Two possible cases are devised in this work: a sensor node is the sink and is in charge of collecting and retransmit, using long range communications, the information retrieved from other sensors on the network (probably consuming energy at a fastest pace), or a mobile sink is available and can move around to help the network use efficiently energy and tackle the hotspot problem. In the former case, it is possible to extend some of the approaches proposed in this work by considering that a sensor can assume this interesting super-role and is used according to the characteristics of the sink, but considering that it may consume energy at a higher rate. In the second case, and depending on the sinks possible locations or movements, it can be possible to assume that this is a problem with multiple sinks constrained to have only one active sink at the same time. Then, decomposition approaches can be explored as strategies to find the optimal solution as it was indicated in this manuscript or heuristic similar to the ones introduced in Chapter 4.

General network design problems It was mentioned in Chapters 4 and 5 that solving the associated pricing subproblems in which connectivity is considered was equivalent to solve a Node-Weighted version of the minimum weight Steiner tree problem. This problems is widely known for being harder to solve than its edge-weighted counterpart. It was shown in this work that it is possible to solve efficiently those problems either by using heuristic or exact approaches. A generalized version of the problem having both weight on edges and on nodes can be devised as a new interesting problem that might have significant applications on the design of communication or electric transmission networks. Additionally, by providing efficient methods to solve this problem it can be possible to consider general versions of the maximum lifetime problem in WSN in which energy consumption is not only associated to the use of certain nodes but also to each connection established.

As one can see, the potential extensions of WSN are numerous, and the limitations are only one's mind limits. As a final word, we still believe that there are plenty of work to do in this area and enough work for several PhD. students that I would love to supervise in the future.

References

- Abdulla, A. E., Nishiyama, H., and Kato, N. (2012). Extending the lifetime of wireless sensor networks: A hybrid routing algorithm. *Computer Communications*, 35(9):1056–1063.
- Abrams, Z., Goel, A., and Plotkin, S. (2004). Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 424–432. ACM.
- Agrawal, A., Klein, P., and Ravi, R. (1991). When trees collide: an approximation algorithm for the generalized Steiner problem on networks. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, STOC '91, pages 134–144, New York, NY, USA. ACM.
- Ahn, N. and Park, S. (2011). A new mathematical formulation and a heuristic for the maximum disjoint set covers problem to improve the lifetime of the wireless sensor network. *Adhoc & Sensor Wireless Networks*, 13:209–225.
- Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349.
- Alfandari, L., Nagih, A., Plateau, A., and Sadki, J. (2013). Hybrid column generation for large-size Covering Integer Programs: application to Transportation Planning. *Computers & Operations Research*, 40(8):1938–1946.
- Alfieri, A., Bianco, A., Brandimarte, P., and Chiasserini, C. (2007). Maximizing system lifetime in wireless sensor networks. *European Journal of Operational Research*, 181(1):390–402.
- Amac Guvensan, M. and Gokhan Yavuz, a. (2011). On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks*, 9(7):1238–1255.
- Amor, H., Desrosiers, J., and Frangioni, A. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184.
- Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568.
- Angelopoulos, S. (2006). The node-weighted Steiner problem in graphs of restricted node weights. In *10th Scandinavian Workshop on Algorithm Theory*, volume 10th Scandinavian Workshop on Algorithm Theory of *Lecture notes in computer science*, pages 208–219, Riga, Latvia.
- Arampatzis, T., Lygeros, J., and Manesis, S. (2005). A survey of applications of wireless sensors and wireless sensor networks. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 719–724, Nicosia, Cyprus. IEEE.

- Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 14–21, Hillsdale, NJ, USA. L. Erlbaum Associates Inc.
- Ben Amor, H., Desrosiers, J., and Valério de Carvalho, J. M. (2006). Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54(3):454–463.
- Benchimol, P., Desaulniers, G., and Desrosiers, J. (2012). Stabilized dynamic constraint aggregation for solving set partitioning problems. *European Journal of Operational Research*, 223:360–371.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.
- Berman, P., Calinescu, G., Shah, C., and Zelikovsky, A. (2005). Efficient energy management in sensor networks. In *Ad Hoc and Sensor Networks*. Nova Science Publisher.
- Biagioni, E. S. and Bridges, K. (2002). The application of remote sensor technology to assist the recovery of rare and endangered species. *International Journal of High Performance Computing Applications*, 16(3):315–324.
- Blum, C., Aguilera, M. J. B., Roli, A., and Sampels, M., editors (2008). *Hybrid Metaheuristics*, volume 114 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Blum, C., Puchinger, J., Raidl, G., and Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135–4151.
- Bokareva, T., Hu, W., Kanhere, S., Ristic, B., Gordon, N., Bessell, T., Rutten, M., and Jha, S. (2006). Wireless sensor networks for battlefield surveillance. In *Proceedings of the land warfare conference*, Brisbane, Australia.
- Briant, O., Lemaréchal, C., Meurdesoif, P., Michel, S., Perrot, N., and Vanderbeck, F. (2008). Comparison of bundle and classical column generation. *Math. Program.*, 113(2):299–344.
- Cardei, I. and Cardei, M. (2008a). Energy efficient connected coverage in wireless sensor networks. *International Journal on Sensor Networks*, 3(3):201–210.
- Cardei, I. and Cardei, M. (2008b). Energy-efficient connected-coverage in wireless sensor networks. *International Journal of Sensor Networks*, 3(3):201–210.
- Cardei, M. and Du, D.-Z. (2005). Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340.
- Cardei, M., Thai, M., Li, Y., and Wu, W. (2005). Energy-efficient target coverage in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1976–1984, Miami, USA. IEEE.

- Cardei, M. and Wu, J. (2004). Coverage in wireless sensor networks. *Handbook of Sensor Networks*, pages 422–433.
- Cardei, M., Wu, J., and Lu, M. (2006). Improving network lifetime using sensors with adjustable sensing ranges. *International Journal of Sensor Networks*, 1(1):41–49.
- Castaño, F., Rossi, A., Sevaux, M., and Velasco, N. (2013). On the use of multiple sinks to extend the lifetime in connected wireless sensor networks. *Electronic Notes in Discrete Mathematics*, 41:77–84.
- Castaño, F., Rossi, A., Sevaux, M., and Velasco, N. (2014). A column generation approach to extend lifetime in wireless sensor networks with coverage and connectivity constraints. *Computers & Operations Research*, 52(B):220–230.
- Cerulli, R., De Donato, R., and Raiconi, A. (2012). Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges. *European Journal of Operational Research*, 220(1):58–66.
- Chen, C. P., Chuang, C. L., Lin, T. S., Lee, C. Y., and Jiang, J. A. (2010). A coverage-guaranteed algorithm to improve network lifetime of wireless sensor networks. *Procedia Engineering*, 5:192–195.
- Chen, D., Liu, Z., Wang, L., Dou, M., Chen, J., and Li, H. (2013). Natural disaster monitoring with wireless sensor networks: A case study of data-intensive applications upon low-cost scalable systems. *Mobile Networks and Applications*, 18(5):651–663.
- Cheng, M., Ruan, L., and Wu, W. (2005). Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2638–2645, Miami, USA. IEEE.
- Chvátal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235.
- Dantzig, G. B. (1963). *Linear programming and extensions*. RAND corporation.
- Dargie, W. and Poellabauer, C. (2010). *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wireless Communications and Mobile Computing.
- de Camargo, R. S., Jr., G. M., and Luna, H. P. (2008). Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & OR*, 35(4):1047–1064.
- de Sá, E. M., de Camargo, R. S., and de Miranda Jr., G. (2013). An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2):185–202.

- Deng, X., Yu, J., Yu, D., and Chen, C. (2012). Transforming area coverage to target coverage to maintain coverage and connectivity for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2012:1–12.
- Deschinkel, K. (2011). A column generation based heuristic for maximum lifetime coverage in wireless sensor networks. In *SENSORCOMM'11, 5-th Int. Conf. on Sensor Technologies and Applications*, pages 209–214, Nice, France. Best papers award.
- Desrochers, M., Desrosiers, J., and Solomon, M. (342-354). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2).
- Deying, L. and Liu, H. (2009). *Wireless Networks: Research, Technology and Applications*, chapter Sensor Coverage in Wireless Sensor Networks, pages 3–31. Nova publishers.
- Dhawan, A., Vu, C. T., Zelikovsky, A., Li, Y., and Prasad, S. K. (2006). Maximum lifetime of sensor networks with adjustable sensing range. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on*, pages 285–289. IEEE.
- Diamond, S. and Ceruti, M. (2007). Application of wireless sensor network to military information integration. In *2007 5th IEEE International Conference on Industrial Informatics*, volume 1, pages 317–322, Vienna, Austria.
- Dietrich, I. and Dressler, F. (2009). On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):1–38.
- Dooms, G., Deville, Y., and Dupont, P. (2005). CP(graph): Introducing a graph computation domain in constraint programming. In *Principles and Practice of Constraint Programming-CP 2005*, pages 211–225. Springer.
- Du Merle, O., Villeneuve, D., Desrosiers, J., and Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237.
- Easwaran, G. and Üster, H. (2009). Tabu search and Benders decomposition approaches for a capacitated closed-loop supply chain network design problem. *Transportation Science*, 43(3):301–320.
- Elhallaoui, I., Villeneuve, D., Soumis, F., and Desaulniers, G. (2008). Dynamic aggregation of set partitioning constraints in column generation. *Operations Research*, 53(4):632–645.
- Errico, F., Crainic, T. G., Malucelli, F., and Nonato, M. (2012). A Benders decomposition approach for the symmetric TSP with generalized latency. Technical Report CIRRELT-2012-78, Montréal, Canada.
- Fages, J. G. and Lorca, X. (2011). Revisiting the tree constraint. In *Principles and Practice of Constraint Programming-CP 2011*, pages 271–285. Springer.

- Farahani, S. (2008). Chapter 6 - battery life analysis. In Farahani, S., editor, *ZigBee Wireless Networks and Transceivers*, pages 207 – 224. Newnes, Burlington.
- Feo, T. and Resende, M. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133.
- Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345.
- Gallais, A., Carle, J., Simplot-Ryl, D., and Stojmenovic, I. (2008). Localized sensor area coverage with low communication overhead. *Mobile Computing, IEEE Transactions on*, 7(5):661–672.
- Gaura, E., Girod, L., Brusey, J., Allen, M., and Challen, G., editors (2010a). *Wireless sensor network: Deployments and Design Framework*. Springer.
- Gaura, E., Girod, L., Brusey, J., Allen, M., and Challen, G. (2010b). Wireless sensor networks: Deployments and design frameworks. *Computer Networks*, 52(12):2292–2330.
- Gentili, M. and Raiconi, A. (2013). α -coverage to extend network lifetime on wireless sensor networks. *Optimization Letters*, 7(1):157–172.
- Geoffrion, A. M. (1972). Generalized Benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260.
- Gervet, C. (1997). Interval propagation to reason about sets: definition and implementation of a practical language. *Constraints*, 1:191–244.
- Gilmore, P. C. and Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859.
- Gilmore, P. C. and Gomory, R. E. (1963). A linear programming approach to the cutting stock problem-part II. *Operations research*, 11(6):863–888.
- Gu, Y., Ji, Y., Li, J., and Zhao, B. (2009a). QoS-aware target coverage in wireless sensor networks. *Wireless Communications and Mobile Computing*, 9(12):1645–1659.
- Gu, Y., Ji, Y., and Zhao, B. (2009b). Maximize lifetime of heterogeneous wireless sensor networks with joint coverage and connectivity requirement. In *EmbeddedCom09: The Eighth International Conference on Embedded Computing*, pages 226–231, Dalian, China.
- Gu, Y., Li, J., Zhao, B., and Ji, Y. (2009c). Target coverage problem in wireless sensor networks: A column generation based approach. In *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*, pages 486–495. IEEE.
- Gu, Y., Zhao, B.-H., Ji, Y.-S., and Li, J. (2011). Theoretical treatment of target coverage in wireless sensor networks. *Journal of Computer Science and Technology*, 26:117–129.

- Guha, S. and Khuller, S. (1999). Improved methods for approximating node weighted Steiner trees and connected dominating sets. *Information and Computation*, 150(1):57–74.
- Hadjidj, A., Souil, M., Bouabdallah, A., Challal, Y., and Owen, H. (2013). Wireless sensor networks for rehabilitation applications: Challenges and opportunities. *Journal of Network and Computer Applications*, 36(1):1 – 15.
- Hamming, R. (1950). Error detecting and error detection codes. *The Bell System Technical Journal*, 29(2):147–160.
- Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467.
- Henna, S. and Erlebach, T. (2013). Approximating maximum disjoint coverage in wireless sensor networks. In *Ad-hoc, Mobile, and Wireless Network*, pages 148–159. Springer.
- Jia, J., Chen, J., Chang, G., and Wen, Y. (2008). Efficient cover set selection in wireless sensor networks. *Acta Automatica Sinica*, 34(9):1157–1162.
- Jong, K. A. and Spears, W. M. (1992). A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1):1–26.
- Karp, R. M. (1972). *Reducibility among combinatorial problems*. Springer.
- Keskin, M. E., AltÄşnel, I. K., Aras, N., and Ersoy, C. (2014). Wireless sensor network lifetime maximization by optimal sensor deployment, activity scheduling, data routing and sink mobility. *Ad Hoc Networks*, 17:18–36.
- Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., and Turon, M. (2007). Health monitoring of civil infrastructures using wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 254–263. IEEE.
- Klein, P. and Ravi, R. (1995). A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, 19(1):104–115.
- Kumar, S., Lai, T. H., and Arora, A. (2005). Barrier coverage with wireless sensors. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, volume 13, pages 284–298, Cologne, Germany. ACM.
- Lai, C., Ting, C., and Ko, R. (2007). An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3531–3538. IEEE.

- Li, X., Xu, X., Zou, F., Du, H., Wan, P., Wang, Y., and Wu, W. (2009). A PTAS for node-weighted Steiner tree in unit disk graphs. In *Proceedings of the 3rd International Conference on Combinatorial Optimization and Applications, COCOA '09*, pages 36–48, Berlin, Germany.
- Li, Y. and Bartos, R. (2013). A survey of protocols for intermittently connected delay-tolerant wireless sensor networks. *Journal of Network and Computer Applications*, 41:411–423.
- Li, Y., Vu, C., Ai, C., Chen, G., and Zhao, Y. (2011). Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(4):695–703.
- Liu, H., Chu, X., Leung, Y., Jia, X., and Wan, P.-J. (2011). General maximal lifetime sensor-target surveillance problem and its solution. *IEEE Transactions on Parallel and Distributed Systems*, 22(10):1757–1765.
- Liu, H., Chu, X., Leung, Y.-W., Jia, X., and Wan, P.-J. (2009). Maximizing lifetime of sensor-target surveillance in wireless sensor networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, Honolulu, Hawaii, USA. IEEE.
- Liu, H., Jia, X., Wan, P.-J., Yi, C.-W., Makki, S. K., and Pissinou, N. (2007). Maximizing lifetime of sensor surveillance systems. *IEEE/ACM Transactions on Networking*, 15(2):334–345.
- Liu, Y. and Liang, W. (2005). Approximate coverage in wireless sensor networks. *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*, pages 68–75.
- Longbottom, R. (2012). Linpack benchmark results on PCs.
- Lu, M., Wu, J., Cardei, M., and Li, M. (2005). Energy-efficient connected coverage of discrete targets in wireless sensor networks. In *Networking and Mobile Computing*, volume 3619 of *Lecture Notes in Computer Science*, pages 43–52, Zhangjiajie, China.
- Lubbecke, M. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, pages 1007–1023.
- Lübbecke, M. E. (2010). Column generation. *Wiley Encyclopedia of Operations Research and Management Science*.
- Macambira, E., Maculan, N., and de Souza, C. (2005). Integer programming models for the sonet ring assignment problem. Technical Report IC-05-026, Universidade Federal do Rio de Janeiro.
- Mackworth, A. (1977). Consistency in Networks of Relations. *Artificial Intelligence Journal*, 8(1):99–118.

- Maniezzo, V., Stützle, T., and Voß, S. (2010). *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, volume 10 of *Annals of information systems*. Springer.
- Marsten, R. E., Hogan, W. W., and Blankenship, J. (1975). The boxstep method for large-scale optimization. *Operations Research*, 23(3):389–405.
- MIT Technology review (2003). 10 emerging technologies that will change the world online.
- Moungla, N. T., Létocart, L., and Nagih, A. (2010). Solutions diversification in a column generation algorithm. *Algorithmic Operations Research*, 5(2):86–95.
- Neame, P. (2000). *Nonsmooth dual methods in integer programming*. PhD thesis, University of Melbourne, Department of Mathematics and Statistics.
- Othman, M. F. and Shazali, K. (2012). Wireless sensor network applications: A study in environment monitoring system. *Procedia Engineering*, 41:1204–1210.
- Prosser, P. and Unsworth, C. (2006). Rooted tree and spanning tree constraints. In *17th ECAI Workshop on Modelling and Solving Problems with Constraints*, Riva del Garda, Italy.
- Puccinelli, D. and Haenggi, M. (2005). Wireless sensor networks: applications and challenges of ubiquitous sensing. *IEEE Circuits and Systems Magazine*, 5(3):19–31.
- Raghunathan, V., Schurgers, C., and Srivastava, M. (2002). Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50.
- Raiconi, A. and Gentili, M. (2011). Exact and metaheuristic approaches to extend lifetime and maintain connectivity in wireless sensor networks. *Lecture Notes in Computer Science*, 6701:607–619.
- Ramsden, D. (2009). *Optimization approaches to sensor placement problems*. PhD thesis, Rensselaer Polytechnic Institute, New York.
- Rebai, M., Le Berre, M., Hnaien, F., Snoussi, H., and Khoukhi, L. (2014). A branch and bound algorithm for the critical grid coverage problem in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2014:1–9.
- Régin, J. C. (2004). Modeling problems in constraint programming, *available on "www.programming.compeoplereginpapersmodelincp.pdf"*. volume 4.
- Régin, J. C. (2011). Global constraints: A survey. In van Hentenryck, P. and Milano, M., editors, *Hybrid Optimization*, volume 45 of *Springer Optimization and Its Applications*, pages 63–134. Springer New York.

- Régin, J. C., Rousseau, L. M., Rueher, M., and van Hoes, W.-J. (2010). The weighted spanning tree constraint revisited. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 287–291. Springer, Cork, Ireland.
- Rossi, A., Singh, A., and Sevaux, M. (2012a). Column generation algorithm for sensor coverage scheduling under bandwidth constraints. *Networks*, 60(3):141–154.
- Rossi, A., Singh, A., and Sevaux, M. (2012b). An exact approach for maximizing the lifetime of sensor networks with adjustable sensing ranges. *Computers and Operations Research*, 39(12):3166–3176.
- Rossi, A., Singh, A., and Sevaux, M. (2013). Lifetime maximization in wireless directional sensor networks. *European Journal of Operational Research*, 231(1):229–241.
- Rossi, F., van Beek, P., and Walsh, T., editors (2006). *Handbook of Constraint Programming*. Foundations of Artificial Intelligence. Elsevier Science Publishers, Amsterdam, The Netherlands.
- Rousseau, L., Gendreau, M., and Feillet, D. (2007). Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668.
- Saharidis, G. K., Boile, M., and Theofanis, S. (2011). Initialization of the Benders master problem using valid inequalities applied to fixed-charge network problems. *Expert Systems with Applications*, 38(6):6627–6636.
- Santos, A. C., Duhamel, C., Belisário, L. S., and Guedes, L. M. (2012). Strategies for designing energy-efficient clusters-based wsn topologies. *Journal of Heuristics*, 18(4):657–675.
- Sha, K. and Shi, W. (2005). Modeling the lifetime of wireless sensor networks. *Sensor Letters*, 3(2):126–135.
- Shi, Y. (2007). *Algorithms and Optimization for Wireless Networks*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA.
- Singh, A., Rossi, A., and Sevaux, M. (2013). Matheuristic approaches for Q-coverage problem versions in wireless sensor networks. *Engineering Optimization*, 45(5):609–626.
- Slijepcevic, S. and Potkonjak, M. (2002). Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 472–476. IEEE.
- Sohraby, K., Minoli, D., and Znati, T. (2007). *Wireless sensor networks: technology, protocols, and applications*.

- Spears, W. (1991). On the virtues of parameterized uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236, La Jolla, CA.
- Talbi, E. G., editor (2013). *Hybrid Metaheuristics*, volume 434 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Tan, H., Hao, X., Wang, Y., Lau, F., and Lv, Y. (2013). An approximate approach for area coverage in wireless sensor networks. *Procedia Computer Science*, 19:240–247.
- Tech investor news (2014). Report: Apple snapping up biomedical, sensor experts in potential wearables push online.
- Tech investor news (2013). Wireless sensor networks: Market shares, strategies, and forecasts, worldwide, 2013 to 2019 online.
- Thai, M. T., Wang, F., Du, D. H., and Jia, X. (2008). Coverage problems in wireless sensor networks: designs and analysis. *International Journal of Sensor Networks*, 3(3):191–200.
- The wall street journal: market watch (2014). Iot market research reports: Technology, application, m2m & semiconductor wireless sensor internet of things 2014 analysis and 2020 forecasts online.
- Tian, D. and Georganas, N. (2005). Connectivity maintenance and coverage preservation in wireless sensor networks. *Ad Hoc Networks*, 3(3):744–761.
- Ting, C. K. and Liao, C.-C. (2010). A memetic algorithm for extending wireless sensor network lifetime. *Information Sciences*, 180(24):4818–4833.
- Tran, V., Reinelt, G., and Bock, H. (2006). Boxstep methods for crew pairing problems. *Optimization and Engineering*, 7(1):33–46.
- Vanderbeck, F. (2005). Implementing mixed integer column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, pages 331–358. Springer US. 10.1007/0-387-25486-2_12.
- Wang, C., Thai, M., Li, Y., Wang, F., and Wu, W. (2009). Optimization scheme for sensor coverage scheduling with bandwidth constraints. *Optimization Letters*, 3(1):63–75.
- Wang, L. and Kulkarni, S. S. (2006). Sacrificing a little coverage can substantially increase network lifetime. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006*, volume 1, pages 326–335, Reston, Virginia, USA. IEEE.
- Warshall, S. (1962). A theorem on boolean matrices. *Journal of the ACM (JACM)*, 9(1):11–12.

- Wentges, P. (1997). Weighted dantzig–wolfe decomposition for linear mixed-integer programming. *International Transactions in Operational Research*, 4(2):151–162.
- Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., and Welsh, M. (2006). Deploying a wireless sensor network on an active volcano. *Internet Computing*, 10(2):18–25.
- Westerlund, A. (2005). *Accelerating Column Generation Schemes*. PhD thesis, Linköpings universitet, Linköpings, Sweden.
- Westerlund, A., Göthe-Lundgren, M., and Larsson, T. (2006). A stabilized column generation scheme for the traveling salesman subtour problem. *Discrete Applied Mathematics*, 154(15):2212–2238.
- Yang, S., Dai, F., Cardei, M., Wu, J., and Patterson, F. (2006). On connected multiple point coverage in wireless sensor networks. *International Journal of Wireless Information Networks*, 13(4):289–301.
- Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330.
- Younis, M., Senturk, I., Akkaya, K., Lee, S., and Senel, F. (2013). Topology management techniques for tolerating node failures in wireless sensor networks: A survey. *Computer Networks*, 58:254–283.
- Yu, Z.-j., Wei, J.-m., and Liu, H.-t. (2009). Energy-efficient collaborative target tracking algorithm using cost-reference particle filtering in wireless acoustic sensor networks. *The Journal of China Universities of Posts and Telecommunications*, 16(1):9–43.
- Zhang, X. (2012). *Energy conservation and lifetime prolongation schemes for distributed wireless sensor network*. PhD thesis, Bond University, Australia, Robina, Gold Coast, Queensland, Australia.
- Zhao, Q. and Gurusamy, M. (2008). Lifetime maximization for connected target coverage in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(6):1378–1391.
- Zhu, C., Zheng, C., Shu, L., and Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):619–632.
- Zorbas, D. and Douligeris, C. (2010). *Sustainable wireless sensor networks*, chapter Power efficient target coverage in wireless sensor networks, pages 355–371. InTech.
- Zorbas, D., Glynos, D., Kotzanikolaou, P., and Douligeris, C. (2010). Solving coverage problems in wireless sensor networks using cover sets. *Ad Hoc Networks*, 8(4):400–415.

List of Figures

2.1	A simple coverage and connectivity model depicting the sensing and communication ranges	13
2.2	Different models for coverage in WSN	13
2.3	Virtual backbone	14
2.4	Optimal schedule for the operation of a simple network	20
2.5	Energy consumption and remaining energy during sensor lifetime	22
2.6	Overview of the column generation algorithm	23
2.7	Relation between coverage and lifetime in WSN	26
3.1	Tail-of effect.	34
3.2	Unstable behavior of dual variables along successive CG iterations.	35
3.3	Evolution of $ \pi^j - \pi^* $ along the iterations using BoxStep method (BoxStep) and du Merle's method (GC3PS).	42
3.4	Comparison of dual variables behavior without stabilization (CG) and the Neame's method.	44
3.5	Comparison of dual behavior in the CG without stabilization (CG) and the intensification strategy (CG-GA).	49
4.1	Multiphase column generation approach.	59
4.2	Neighborhood structures considered in the VNS approach for PS.	65
4.3	Evolution of the objective function for CMLP.	66
4.4	Dual variables values behavior for several stabilization and intensification strategies.	67
4.5	Evolution of the CG-EXACT (Instance S100_T15_a70)	71
4.6	Comparison of average time spent at each CG phase.	72
5.1	Representation of feasible and infeasible partitions in a simple network.	79
5.2	Column generation approach to solve CMPL-MR	84
5.3	Comparison of objective function evolution throughout CG for instance CMLP_MR_001.	91
6.1	A wireless sensor network providing full and partial coverage	97
6.2	Representation of a valid partition in a simple network.	103
6.3	Reduction of the number iterations (%) by returning several columns throughout CG+EA+CP.	108
6.4	Reduction on the CPU time (%) by returning several columns throughout CG+EA+CP.	109
6.5	Objective function evolution throughout CG iterations.	109

List of Tables

2.1	A comparison of WSN main families	11
3.1	Comparison of the updating strategies in the BoxStep method.	42
3.2	Comparison of updating strategies in the Du Merle’s method.	43
3.3	Comparison of Neame’s method and CG without stabilization.	45
3.4	Evaluation of the effect of α in the number of iterations in the Neame’s method.	45
3.5	Evaluation of the effect of α in the computational time (s) for the Neame’s method.	45
3.6	Comparison of number of iterations for κ -intensification and diversification techniques.	49
3.7	Comparison of number of iterations for κ -intensification and diversification techniques.	51
4.1	Comparison of the proposed approaches on the Raiconi and Gentili set of instances.	68
4.2	Comparison of average lifetime and average running time ($R_s = 100, R_c = 125$).	70
4.3	Comparison of average lifetime and average running time ($R_s = 125, R_c = 125$).	71
5.1	Notations.	77
5.2	Experimental results for CMLP-MR ($E_r = 0.8, E_s = 1.0$).	89
5.3	Experimental results for CMLP ($E_r = E_s = 1.0$).	90
6.1	Number of optimal solutions computed by each method in the different groups of instances	111
6.2	Comparison of computational time (s) required for the proposed approaches	112
6.3	Average network lifetime computed through the proposed approaches	113